AGARD-LS-200

AGARD-LS-200

AGARD

ADVISORY GROUP FOR AEROSPACE RESEARCH & DEVELOPMENT
7 RUE ANCELLE, 92200 NEUILLY-SUR-SEINE, FRANCE

AGARD LECTURE SERIES 200

Knowledge-Based Functions in Aerospace Systems

(Systèmes de guidage et de pilotage aérospatiaux
à base de systèmes experts)

The material in this publication was assembled to support a Lecture Series under the sponsorship of the Mission Systems Panel of AGARD and the Consultant and Exchange Programme of AGARD presented on 6-7 November 1995 in Madrid, Spain, 9-10 November 1995 in Châtillon, France, and 16-17 November 1995 in NASA AMES Research Centre, California, USA.

NORTH ATLANTIC TREATY ORGANIZATION

Published November 1995

Distribution and Availability on Back Cover

AGARD-LS-200

AGARD

ADVISORY GROUP FOR AEROSPACE RESEARCH & DEVELOPMENT

7 RUE ANCELLE, 92200 NEUILLY-SUR-SEINE, FRANCE

AGARD LECTURE SERIES 200

Knowledge-Based Functions in Aerospace Systems

(Systèmes de guidage et de pilotage aérospatiaux à base de systèmes experts)

The material in this publication was assembled to support a Lecture Series under the sponsorship of the Mission Systems Panel of AGARD and the Consultant and Exchange Programme of AGARD presented on 6-7 November 1995 in Madrid, Spain, 9-10 November 1995 in Châtillon, France, and 16-17 November 1995 in NASA AMES Research Centre, California, USA.

North Atlantic Treaty Organization

Organisation du Traité de l'Atlantique Nord

The Mission of AGARD

According to its Charter, the mission of AGARD is to bring together the leading personalities of the NATO nations in the fields of science and technology relating to aerospace for the following purposes:

— Recommending effective ways for the member nations to use their research and development capabilities for the common benefit of the NATO community;

— Providing scientific and technical advice and assistance to the Military Committee in the field of aerospace research and development (with particular regard to its military application);

— Continuously stimulating advances in the aerospace sciences relevant to strengthening the common defence posture;

— Improving the co-operation among member nations in aerospace research and development;

— Exchange of scientific and technical information;

— Providing assistance to member nations for the purpose of increasing their scientific and technical potential;

— Rendering scientific and technical assistance, as requested, to other NATO bodies and to member nations in connection with research and development problems in the aerospace field.

The highest authority within AGARD is the National Delegates Board consisting of officially appointed senior representatives from each member nation. The mission of AGARD is carried out through the Panels which are composed of experts appointed by the National Delegates, the Consultant and Exchange Programme and the Aerospace Applications Studies Programme. The results of AGARD work are reported to the member nations and the NATO Authorities through the AGARD series of publications of which this is one.

Participation in AGARD activities is by invitation only and is normally limited to citizens of the NATO nations.

The content of this publication has been reproduced
directly from material supplied by AGARD or the authors.

Published November 1995

ISBN 92-836-1026-1

Printed by Canada Communication Group
45 Sacré-Cœur Blvd., Hull (Québec), Canada K1A 0S7

Contents

	Page
Abstract/Abrégé	**iv**
List of Authors/Speakers	**v**
Introduction by Prof. Heinz WINTER	**I**
Functional Analysis/Decomposition of Closed-loop, Real-time Work Processes by Dr. Milton B. ADAMS	**1**
Distributed, Collaborative, Knowledge Based Air Campaign Planning by Mr. Mark T. MAYBURY	**2**
Paper 3 — cancelled	
Functional Development and field test of CASSY — a knowledge-based cockpit assistant system by Prof. Reiner ONKEN	**4**
Development environment for Knowledge-Based Systems. Some examples of Application: The "Copilote Electronique" Project by Mr. Gilles CHAMPIGNEUX	**5**
Knowledge-Based Systems Applications for Automating Spacecraft Operations: Users Feedback and Lessons Learned by Mr. Jean-Michel DARROY, Mr. Arnaud DE SAINT VINCENT, Mr. François LECOUAT	**6**
Design Principles and Algorithms for Automated Air Traffic Management by Dr. Heinz ERZBERGER	**7**
Structures, Architectures and Design Principles for Dynamic Planning Functions in ATM by Dr. Uwe VÖLCKERS, Mr. D. BÖHME	**8**
Bibliography	**B**

Abstract

In aerospace systems classical control technology has enabled the transfer of functions of the human operator to machines which need not be based on the explicit evaluation of knowledge. Symbolic data processing, neural networks and the techniques of artificial intelligence now permit the design of automatic systems which can explicity make use of knowledge stored in computers.

The Lecture Series presents a conceptual framework for the automation of knowledge-based control and management functions in aerospace systems, which are usually carried out by human operators. It describes the structure of these functions, discusses successful examples of application and gives recommendations for further studies. The detailed discussion of the application examples, together with the experiences and lessons learned from these implementations will help potential builders of knowledge-based systems for aerospace applications to learn from the experts in this field.

This Lecture Series, sponsored by the Mission Systems Panel of AGARD, has been implemented by the Consultant and Exchange Programme.

Abrégé

La mise en œuvre des technologies de contrôle classiques dans les systèmes aérospatiaux a permis le transfert des fonctions exécutées par l'opérateur humain à des machines qui ne sont pas nécessairement basées sur l'évaluation explicite des connaissances. Le traitement symbolique des données, les réseaux neuronaux et les techniques de l'intelligence artificielle permettent désormais de concevoir des systèmes automatiques capables de tirer parti des connaissances stockées dans les ordinateurs.

Ce cycle de conférences présente un cadre conceptuel pour l'automatisation des fonctions de contrôle et de gestion à base de connaissances dans les systèmes aérospatiaux, qui sont normalement exécutées par des opérateurs humains. Il donne la description de ces fonctions, présente quelques exemples d'applications réussies et fait des recommandations concernant de futures études. La discussion approfondie des exemples d'applications, ainsi que l'expérience et les enseignements à tirer de ces réalisations permettra aux constructeurs potentiels de systèmes à base de connaissances pour des applications aérospatiales de se renseigner directement auprès des experts dans ce domaine.

Ce cycle de conférences est présenté dans le cadre du programme des consultants et des échanges, sous l'égide du Panel de systèmes de mission de l'AGARD.

List of Authors/Speakers

Lecture Series Director: Prof. Heinz WINTER
DLR Institut für Flugführung
Postfach 3267
D38022 Braunschweig
GERMANY

Dr. Milton B. ADAMS
Associate Director of Applied
Information and Automation Systems
The Charles Stark Draper Laboratory, Inc.
555 Technology Square — MS89
Cambridge, MA 02139
USA

Mr. Gilles CHAMPIGNEUX
Dassault Aviation
DGT-DSA
78 Quai Marcel Dassault
BP 300
92214 Saint Cloud
FRANCE

Dr. Heinz ERZBERGER
NASA AMES Research Centre
MS 210-9 Moffett Federal Airfield
CA 94035-1000
USA

Mr. Mark T. MAYBURY
Bedford Artificial Intelligence
Centre
Advanced Information Systems
Technology Department
The MITRE Corporation
Mail Stop K331, 202 Burlington Rd.
Bedford, MA 01730-1420
USA

Prof. Reiner ONKEN
Universitat der Bundeswehr
München
Werner-Heisenberg-Weg 39
D-85577 Neubiberg
GERMANY

Mr. Jean Michel DARROY
Matra Marconi Space
Innovation & Information
Engineering
Division Manager
31 rue des Cosmonautes
Z.I. du Palays
31077 Toulouse Cedex
FRANCE

Dr. Uwe VÖLCKERS
Institute of Flight Guidance
German Aerospace Research Establishment
Lilienthalplatz 7
D-38108 Braunschweig
GERMANY

Co-Authors

Mr. D. BÖHME
Institute of Flight Guidance
German Aerospace Research Establishment
Lilienthalplatz 7
D-38108 Braunschweig
GERMANY

Mr. François LECOUAT
MATRA MARCONI SPACE
Z.I. du Palays
31 rue des Cosmonautes
31077 Toulouse Cedex
FRANCE

Mr. Arnaud de SAINT VINCENT
MATRA MARCONI SPACE
Z.I. du Palays
31 rue des Cosmonautes
31077 Toulouse Cedex
FRANCE

KNOWLEDGE-BASED FUNCTIONS IN AEROSPACE SYSTEMS AN INTRODUCTION

Heinz Winter
DLR Institut für Flugführung
Postfach 3267
D38022 Braunschweig, Germany

1. Objective

Classical control theory has enabled the engineers to transfer such human operator functions to machines (control systems), which require no explicit handling of knowledge. The advent of symbolic data processing, neural network and artificial intelligence techniques makes it now possible to design automatic systems also for such functions which make explicit use of knowledge stored in computers.

In a former Working Group of AGARD, Guidance and Control functions in aerospace systems have been analysed with respect to the knowledge-based content of actions carried out by human operators. Such functions are performed, for example, in the cockpit of a military or civilian airplane, at an air traffic controller's work position, at a mission planning work station or in a space vehicle control center. The results of this AGARD activity have been documented in a report [1].

The present Lecture Series concentrates on the discussion of such examples of application where knowledge-based functions have already been successfully automated in aerospace systems. The goal is that the detailed presentation of these projects, together with the discussion of experiences and lessons learned from the implementations shall help potential builders of knowledge-based systems to design similar systems.

2. Knowledge-Based Functions

We can study knowledge-based functions by considering the general structure of human behavior. The goal-directed interactions of man with the surrounding world can be decomposed into the functional elements (subfunctions) of the so-called recognize-act-cycle [2] (or stimulus-response-cycle):

a) MONITORING: Recognize the actual state of the world and compare it with the desired state (which corresponds to the goal of the interaction).
b) DIAGNOSIS: Analyse the deviations of actual and desired state.
c) PLAN GENERATION: Think about actions to modify the state of the world.
d) PLAN SELECTION: Decide about the necessary actions to reach the desired state.
e) PLAN EXECUTION: Take the necessary actions to change the state of the world.

For many simple tasks a person's physical sensors (eyes, ears, etc.), his brain and his physical effectors (arms, legs, etc.) are sufficient to carry out these functions. We call this "manual interaction". More demanding tasks (e.g. flying a military airplane) go beyond the capabilities of his physical sensor/effector equipment. Therefore, man has invented a great variety of tools to support his interactions with the world. The tools may support ("semi-automatic interaction") or even replace the human functions ("fully automatic").

Generally, knowledge-based human functions are required to solve a problem in the surrounding world. In these cases, the information processing carried out by the human brain in order to find a solution of the problem can be described in a similar way by the following chain of functions:

- Recognition of a problem in connection with the actual state of the world and its representation in a "mental model". Definition of the desired goal state.
- Construction of control operations to bring the surrounding world from the recognized problem state to desired goal states.
- Selection of criteria to evaluate the different control strategies.
- "Simulation" of the effect of the control strategies on the world to assess their effectivity.
- Evaluation of the possible control strategies.
- Selection of the appropriate control strategy to "best" drive the surrounding world to the desired goal state.

3. Work Systems

In our industrial society many of the human interactions with the world happen in so-called work systems [3], where man spends a great part of his life. The goal of a work system is to fulfill a certain task, for which it has been built. It normally consists of the elements (see **Figure 1**): Operator, Work Object and the Tool(s). The tools are devices or machines (or

Paper presented at the Mission Systems Panel of AGARD and the Consultant and Exchange Program of AGARD held in Madrid, Spain, on 6-7 November 1995; Châtillon, France, 9-10 November 1995; and NASA AMES Research Centre, California, USA, 16-17 November 1995 and published in LS-200.

sometimes other people) which help the operator to fulfill the task. The system elements interact through the operator- and the work-object-interfaces, with the goal to produce a certain output, the product. The operator can interact with the work-object directly (manual operation) or with the help of a tool (semi-automatic or automatic operation). The declarative (or object-oriented) representation which describes the elements making up the work-system in Figure 1 is instantiated in that figure with the situation of a pilot in the cockpit of an airplane. Here the operator is the pilot, the work-object is the airplane and the tool is the autopilot of the aircraft. The

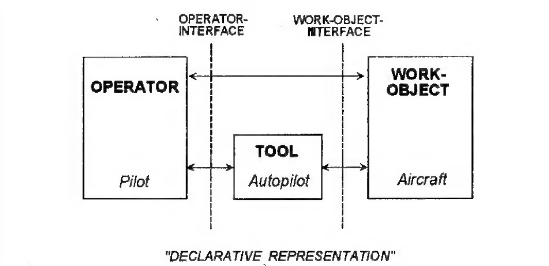


Fig. 1: WORK SYSTEM AIRCRAFT

goal is to fly the airplane in accordance with the flight plan (or the mission plan in the military case) subject to the ground rules of safe flight and possible directives of the ATC.

Another (complementary) way of viewing a work-system is the procedural (or function-oriented) representation in **Figure 2**, which describes those functions performed by the system elements that are required in order to obtain the product.

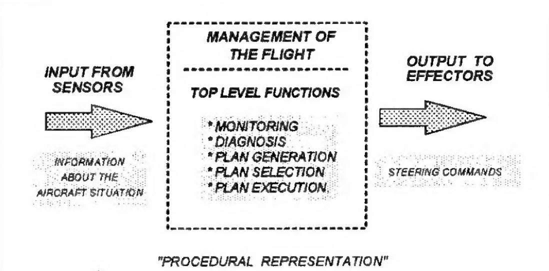


Fig.2: FLYING AN AIRCRAFT AS A WORK PROCESS

We can describe the top level functions also in this case as

- Monitoring
- Diagnosis
- Plan generation
- Plan selection
- Plan execution.

In manual flight, the pilot transforms the aircraft state into its desired value, feeding the output of the work-process (the control commands) to the effectors (the actuators of the airplane). In the case of a semi-automatic or automatic flight, tools (like the autopilot) contribute to performing (partially or totally) the top level functions.

The complex aerospace systems which are discussed in this Lecture Series (e.g. a space mission control center, air traffic management systems) can be represented as networks of elementary work systems.

4. Man-Machine Interactions

A framework has been described by J. Rasmussen [4], which can be used for a better understanding of the interactions of a human operator with a tool or a work-object. Following his ideas, these interactions can take place on three levels:

- Skill-based activities which are carried out almost subconsciously and automatically. A skilled operator has a large repertoire of automated sensory-motor subroutines which he can put together to form larger patterns of behavior in order to interact with the tool or the work-object.
- Rule-based activities steered by stored rules which have been learned during instruction or by experience. These rules cover all routine situations for which there are no automated sensory-motor subroutines.
- Knowledge-based activities which take place in non-routine situations, where no learned rules or skills can solve the problem. In such situations the operator has to develop new problem solutions based on his objectives and knowledge about the work-object and the world.

The following **Table** describes some of the characteristics of these interaction levels.

Table: Characteristics of the levels of interaction between a human or a tool and a work-object

Interaction Levels	*Required Response Time*	*Structure of Info. Processing in the Brain*	*Structure of Info. Processing in a computer*
Skills	*short*	*connectionist*	*neural network*
Rules	*medium*	*cognitive*	*rule-based system*
Knowledge	*long*	*cognitive*	*knowledge-based syst.*

In analogy to models of cognitive science one can also assume that skills are stored in the human brain as

"situation/action" patterns, rules in the form of symbolic "if(situation)-than(action)" pairs. Knowledge can be represented in declarative form (knowledge about the world, the work-object, the tools and the human operator) or in procedural form (knowledge about actions and their use for problem solving).

When building a machine, which performs skill-, rule- or knowledge-based functions, an engineer may use computer programs of similar structure, as indicated in the Table.

5. Functional Architecture

Most of the examples discussed in this Lecture Series are related to the management of aerospace systems. Based on the results of the Working Group [1], the general structure of such management functions can be described as shown in the **Figure 3**.

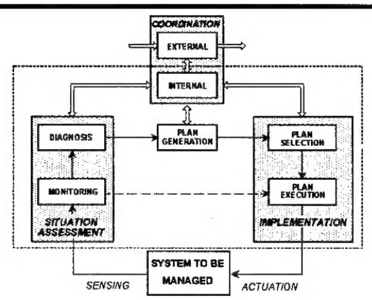


Fig. 3: STRUCTURE OF THE MANAGEMENT FUNCTION

The functional elements of the management function are arranged in a certain functional architecture, and they have been grouped together in the more general functions

- situation assessment
- plan generation
- plan implementation, and
- coordination.

The coordination function in this architecture controls the execution of the individual functional elements, and coordinates the total management function with other work systems. As already mentioned, Figure 3 describes the generalized structure of the management function. The application examples in the Lecture Series will show the functional architectures which have been chosen by the design engineers for the individual examples.

REFERENCES

[1] AGARD-AR-325: *Knowledge-Based Guidance and Control Functions.* AGARD, January 1995.

[2] K.R. Boff, L. Kaufmann and J.P. Thomas (Eds.): *Handbook of Perception and Human Performance.* Vol. I and II. John Wiley and Sons, New York 1986.

B.H. Kantowitz and R.D. Sorkin: *Human Factors: Understanding People-System Relationships.* John Wiley and Sons, New York 1983.

[3] H. Luczak *Arbeitswissenschaft.* Springer Verlag, Berlin, 1993.

Methodenlehre des Arbeitsstudiums. REFA, Verband für Arbeitsstudien und Betriebsorganisation e.V. Hanser Verlag, München 1984.

R. Bokranz und K. Landau: *Einführung in die Arbeitswissenschaft.* UTB 1619, Ulmer Verlag, Stuttgart 1991.

[4] J. Rassmussen: *Skills, rules and knowledge.* IEEE Transactions on Systems, Man, and Cybernetics. Vol. SMC-13, No. 3, May/June 1983.

Functional Analysis / Decomposition of Closed-loop, Real-time Work Processes

Milton B. Adams
The Charles Stark Draper Laboratory, Inc.
555 Technology Square, Mail Station 89
Cambridge, Massachusetts 02139, USA

1. TOP LEVEL FUNCTIONAL STRUCTURE OF LARGE-SCALE, REAL-TIME GUIDANCE AND CONTROL PROCESSES

Performing a functional analysis is the first step in developing a solution to a complex problem. The functional analysis of a generic complex, real-time, closed-loop process presented here serves to establish a common framework and a common language for describing a variety of specific guidance and control related processes.

The two principal components of a functional analysis are

(1) a functional decomposition and

(2) a detailed description of the information flow contained in the interfaces between the individual functions comprising that decomposition.

The approach taken here is the classical structured analysis with a dataflow representation. A dataflow representation has been chosen because it readily lends itself to a hierarchical description. That is, each individual function can be further decomposed into more refined subfunctions which, when taken together, have combined inputs and outputs which are consistent with those of the original function. Neither control flow representations nor object-oriented analysis is addressed.

The functional analysis presented here is not intended to represent the uniquely "correct" decomposition. Indeed, what is presented reflects similarities with descriptions given elsewhere for real-time planning and decision-making systems, [e.g., 1 and 2]. Rather the intent is to develop a description at a sufficiently high level of abstraction that can be used as a template for the functional description of a variety of processes ranging from the hierarchy of decision-making, problem-solving and planning functions required in a battlefield setting or air traffic flow control to the onboard functions required for the planning and execution of a mission of a tactical fighter aircraft.

Ultimately, the objective of the functional analysis is to identify that subset of functions within a real-time, closed loop process which might be designed and implemented via knowledge-based approaches. The analysis establishes a common framework and serves as a point of departure for the more detailed functional descriptions for the systems addressed by the other components of this lecture series.

The internal functionality of processes is defined in a manner that is independent of the mechanisms that ultimately will be used for their implementation: i.e., man vs. machine, hardware vs. software. Allocation of function to person or machine or to a person and machine in concert is not addressed. Indeed, in so-called "human centered" designs [1 - 3] or "mixed-initiative" systems [1 - 4], the extent of human participation in each function is dynamic and can range from totally manual to completely autonomous.

1.1 Functional Decomposition: Generic

Our presentation of the functional analysis of a generic closed-loop decision making system begins with a description of the basic elements that are depicted in Figure 1.1. A more detailed functional breakdown follows this generic system description and includes an overview of each of the decision-making functions contained in that more detailed decomposition.

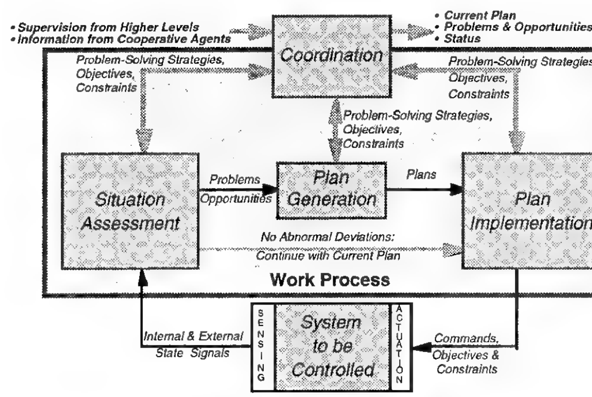


Figure 1.1 Functional Decomposition: Basic Sub-Functions of a Generic Process

1.1.1 Basic Process Sub-Functions Comprising the Generic Decomposition

The generic decomposition of a real-time, closed-loop process illustrated in Figure 1.1 comprises a set of individual sub-functions including: *Coordination*, *Situation Assessment*, *Plan Generation*, and *Plan Implementation*. Coordination translates external inputs into problem-solving strategies, objectives and constraints that are to be employed by the other sub-functions. The Plan Generation and Plan Implementation functions do exactly as their names suggest. Situation Assessment monitors the state of the world both to determine whether the objectives and constraints embodied in the current operational plan/solution[1] are being honored as well as to detect previously unforeseen opportunities that may allow

1 The terms plan and solution are often used interchangeably throughout this section.

Paper presented at the Mission Systems Panel of AGARD and the Consultant and Exchange Program of AGARD held in Madrid, Spain, on 6-7 November 1995; Châtillon, France, 9-10 November 1995; and NASA AMES Research Centre, California, USA, 16-17 November 1995 and published in LS-200.

the accomplishment of more goals or objectives than those embodied in the current plan. The result of this monitoring can be either a decision to continue to pursue the current plan or to create a new plan in order to accommodate problems or to take advantage of opportunities. Note again that each of these functions is influenced by the output of the Coordination function as well as signals/data/information that are internal to the Assess-Plan-Implement loop shown in Figure 1.1.

The decomposition of the generic process illustrated in Figure 1.1 provides a framework for the more detailed discussions that follow. Indeed, within this framework one is capable of describing the functionality of closed-loop control systems ranging from classical closed-loop control systems to real-time hierarchical problem-solving such as battlefield management. Before, addressing the application of this analysis to those types of problems, we will expand slightly on this initial view of the generic functional analysis.

1.1.2 Inputs and Outputs of a Sub-Function

In the process of formulating a plan / decision / solution, a sub-function requires several classes of input information. These classes include the problem-specific, real-time data and signals describing the elements of the current state of the world that are relevant to the task of the sub-function shown entering on the left in Figure 1.2. In addition, tasking and strategy *control* inputs are required to define problem-solving criteria such as time available to generate a solution, the overall objective function, costs and constraints. These control inputs are influenced by inputs from a higher level entity and are shown entering at the top of the figure. Finally, in order to generate a solution, *knowledge* inputs including quasi-static, a priori information regarding the state of the world such as maps as well as problem-solving mechanisms that may be employed e.g., heuristics, search algorithms and inferencing techniques must be known. The knowledge inputs are shown entering at the bottom of the figure.

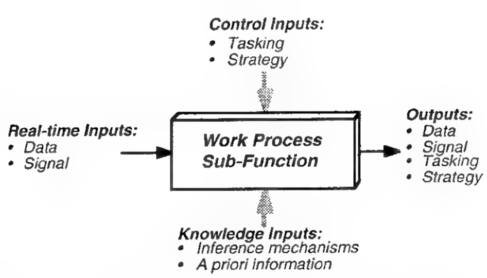


Figure 1.2 Generic Sub-Function: Inputs and Outputs

The standard format used in depicting processes in the ensuing sections shows the *control* inputs with gray arrows but does not explicitly show the *knowledge* inputs. This does not imply that *knowledge* inputs are considered to be unimportant nor that they should be overlooked, rather we have chosen only to represent explicitly the dataflow for the more dynamic, real-time information.

The outputs representing the results of the decision-making can potentially include any or all of the classes of data and information that have been described above as inputs. These outputs, of course, serve as inputs to other sub-functions of the overall closed-loop process (e.g., as shown in Figure 1.1).

1.1.3 More Detailed Functional Decomposition

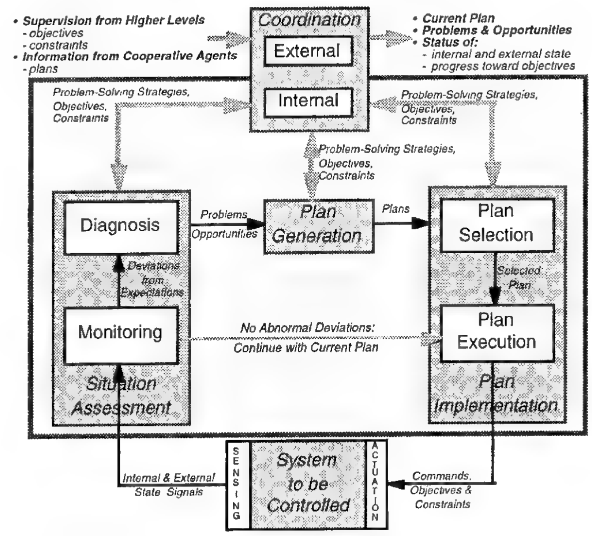


Figure 1.3a Detailed Decomposition

In order to more clearly describe the functions that are executed during normal operation of the real-teim process, the Situation Assessment and Plan Implementation functions have been further decomposed into *Diagnosis* and *Monitoring* and *Plan Selection* and *Plan Execution*, respectively.

This further decomposition makes it explicit that Diagnosis, Plan Generation and Plan Selection are required within the real-time loop only when Monitoring detects a significant deviation from the expected situation. These, along with a more detailed view of the Coordination function, are shown in Figure 1.3a. In particular, the Coordination function is further decomposed to show both that it must organize and execute communications, when appropriate, with *external* agents and that it must continuously provide *internal* coordination to influence and control the execution of the sub-functions within the closed-loop process.

In order to make the description of the elements of the generic functional decomposition more concrete, that description is elaborated in the context of an abstraction of an aircraft Guidance and Control process and its inputs illustrated in Figure 1.3b.

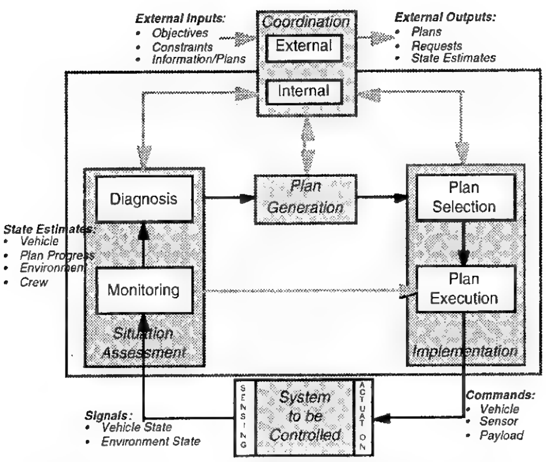


Figure 1.3b Functional Decomposition of a Knowledge-Based Guidance and Control System

1.1.4 Sub-Functions within the Closed-loop Process

The discussion of a generic sub-function presented earlier in Section 1.1 defines the classes of inputs and outputs for each function and thereby serves as a basis for describing the inputs and outputs of the specific, individual sub-functions shown in Figure 1.3a. The discussions below focus on information and the signals flowing within the real-time, decision-making loop. The description of each individual sub-function is, in some cases, contained implicitly in the description of its inputs and outputs, i.e., the function transforms its inputs into its outputs.

Situation Assessment

Monitoring

Inputs: Signals containing information about both the internal and external states of the system, i.e., the internal health and status of the *system-to-be-controlled* and the elements of the state of the environment external to the system-to-be-controlled that may influence the decision-making process, are provided as inputs from sensing systems. Problem-solving strategies that input include measures for and characterization of what constitutes a "significant" deviation from the expected state of the system. These are used to determine whether the normal operating loop must be augmented to include Diagnosis, Plan Generation and Plan Selection.

Outputs: The output of the monitoring function is its decision indicating that either there are no abnormal deviations or there are significant deviations from expectations. In the former case, control is passed to the Plan Execution function. In the latter case, control is passed to the Diagnosis function along with an indication of the nature of the deviations. Although not shown explicitly in Figure 1.3, the state estimates (and uncertainties in those estimates) employed and/or generated by the Monitoring function are made globally available to all of the individual sub-functions.

Diagnosis

Inputs: The principal input is the output of the monitoring function: an indication of a deviation from expectations.

Output: In the case that a problem has been detected by the monitoring function, the output of Diagnosis is a determination of the source of the problem and a characterization of the impact of that problem on the capabilities of the system-to-be-controlled and/or on the ability of the system to pursue the current plan (e.g., an inability to achieve current or future objectives contained in the plan). In the case that the deviation from expectations represents a potential opportunity, the diagnosis function must characterize that opportunity in terms that will allow the Plan Generation function to incorporate any attendant new goals or objectives into the plan of activities to be pursued. Diagnostic information, similar to state estimates produced by the Monitoring function, are made globally available.

Plan Generation

Inputs: A new plan may be required in response to either detected and diagnosed problems or unexpected opportunities that are provided as input from the Diagnosis function. The Plan Generation function may create a variety of plans that trade off among different levels of constraint and/or different objective functions (i.e., plan optimization criteria). The objective function(s) and constraints are provided by the Coordination function. Furthermore, a variety of algorithms or search techniques may be applied in generating plans. These algorithms and search techniques are knowledge inputs. The decision as to which plan generation mechanism(s) to employ is made as a function of the strategy component of the control inputs from the Coordination function. For example, a quick heuristic may be required if a new plan is needed immediately to accommodate a serious (potentially mission or safety critical) problem that has been diagnosed. In other situations, it may be acceptable to continue to pursue the current plan while a more considered search of the plan space is executed in an attempt to refine the current solution to include additional opportunities or to accommodate minor degradations in the capabilities of the system-to-be-controlled.

Outputs: The output is the plan or set of plans that has been generated along with the figures of merit for (i.e., the values of) those plans and the resources required for their execution.

Plan Implementation

Plan Selection

Inputs: Given the plans that have been generated by the Plan Generation function, the Plan Selection function must choose from among those plans the

one that best achieves the overall objectives as provided by the coordination function. For instance, there may be multiple objective functions, and given a strategy for selecting among a set of plans, the Plan Selection function must make the choice of a single plan.

Indeed, a change in plan may not be warranted if there is no plan in the set of generated plans whose value sufficiently exceeds the value of the current plan[2]. The interpretation of "sufficiently exceeds" is made in the context of strategy inputs from the coordination function.

Output: The selected plan is the output and it is made globally available to all sub-functions.

Plan Execution

Inputs: Control is passed to Plan Execution either from Plan Selection, in the case when a new plan is generated and selected, or from Monitoring, in the case when no abnormal deviations are detected. In the former case, the current plan is updated by splicing the new plan onto the current plan at a designated point in time (either immediately or at some point in the future). The "splice time-point" and associated expected state of the system at the splice time are contained as elements of the new plan.

Given the current state of the system, Plan Execution interprets the current (or updated current) plan and creates the commands for the system-to-be-controlled. Execution of these "set-point" commands by the system-to-be-controlled results in the pursuit of the current plan.

Note that even when there are no detected abnormal deviations, under normal operations there typically will be minor deviations from the expected state. Thus, in addition to creating set-point commands for the pursuit of the current plan, an auxiliary role of the Plan Execution function is to determine "perturbation" commands that will correct for the range of normally expected deviations of the actual system state from the planned state.

Outputs: Commands in the form of both objectives and constraints for the system-to-be-controlled are the output of the Plan Execution function.

Coordination

The performance of the Internal and External Coordination functions are key to the overall performance of the process (Especially, when the process under consideration is embedded within a hierarchy of processes as discussed later). Concomitantly, their functional analysis and design are probably the most challenging (and, unfortunately, the most often neglected or overlooked). If allocated to a machine (i.e., if automated), internal and external coordination represent a significant potential for solution by and challenge for Knowledge Based technologies.

External Coordination

The External Coordination sub-function is responsible for receiving and analyzing inputs from external agents: those external agents may be either higher level supervisory agents or cooperative agents solving problems or making decisions at the same level. External Coordination is also responsible for assembling and transmitting information to those same agents. Here "assembling" implies deciding:

(1) What to transmit,

(2) When to transmit it and

(3) To whom to transmit.

Examples of interactions with external agents:

(1) Decide what gathered and assessed information to communicate

(2) Determine what decisions/plans to communicate

(3) Request assistance from supporting agents

(4) Interpret requirements from higher planning authorities

(5) Negotiate with subordinate and collateral planning levels

Inputs: Inputs include supervisory commands or information from higher levels in the form of objectives and constraints as well as plans or solutions generated by other agents at the same level of problem solving. In addition, information regarding elements of the state of the world that are relevant to the problem to be solved are also received.

Outputs: The outputs include the current plan or solution, the status of the internal and external states and progress toward the accomplishment of the objectives in the current plan or solution. In addition, any diagnosed problems or opportunities that may be of interest to external agents are also included as outputs.

Internal Coordination

The principal function of Internal Coordination is to develop criteria and strategies for controlling or guiding the real-time decision-making performed by other functions within the real-time process. By monitoring the assessed situation including progress toward the current solution and the state of both the system-to-be-controlled and the external environment and by taking into consideration any plans developed by other agents and objectives and constraints input from higher level authorities, Internal Coordination develops strategies for controlling the other individual sub-functions including providing:

(1) The criteria for deciding when replanning is required,

2 A re-evaluation of the current plan is required in the face of any diagnosed problems with either the system to be controlled or the external environment within which that system must operate

(2) The time allocated to generating a solution and

(3) Cost/objective functions and constraints to be employed in generating a solution.

1.2 Hierarchical Decomposition

In order to make the solution of complex problems tractable, they are often decomposed into simpler, decoupled subproblems that can be solved (nearly) independently [5 - 10]. If the decomposition is formulated with proper coordination across the processes generating the solutions to the subproblems, then the set of solutions for the subproblems can be combined into a near-optimal, complete solution for the original, more complex problem. The coordination of the solutions to the subproblems is managed by a *Master Problem Solving Level as* shown in Figure 5.

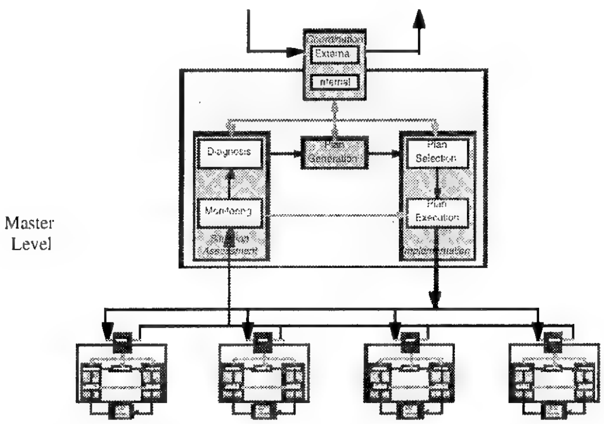


Figure 1.5 Hierarchical View: Distributed Problem-Solving at the Lower Level

The selection of the individual subproblems to be addressed at each level, the specification of their local performance criteria and the modeling of their interactions are all part of an integrated approach to developing a hierarchical decomposition. An important goal in these selections is to maintain a balance in the complexity of decision-making effort across all levels of the hierarchy. The following basic questions must be answered when designing a hierarchy:

- How many levels are required?
- How should problem-solving be partitioned across levels?
- What constraints and objectives should be passed from level to level?
- What happens when a level cannot meet its objectives and/or honor its constraints?

Analytical approaches to decompositions of *Large Scale Optimization Problems* and the essential role played by the subproblem coordination function (which is itself formulated as an optimization problem) have been developed over the last three decades [5, 6]. These approaches have been extended in the development of methodologies for the decomposition of *Large Scale Control Problems* [7, 8]. These formal analytical developments help to establish methodologies for achieving decompositions wherein the subproblems are properly coordinated via a higher, Master, level.

Two types of problems which are amenable to hierarchical decompositions are described in the following: (1) Spatially and functionally distributed problems and (2) Temporal planning problems. Often, decompositions exhibit a mix of temporal and spatial character.

1.2.1 *Decomposition for Spatially Distributed Problem-Solving:*

A hierarchical decomposition can be viewed as a recursive implementation of the functional decomposition described in Sections 1.1 and 1.2 wherein the "system-to-be-controlled" is one or more lower level processes that are "controlled" or coordinated by an upper Master level as illustrated in Figure 1.5. Note that each of the lower level processes has identical structure to that of the upper level and, as we will see later, may have its "system-to-be-controlled" further decomposed into a set of processes at a yet lower level. A natural form of decomposition is one wherein there is a physical or geographic distribution of the entities at the lower level. The nature and implementation of the decomposition is strongly influenced by the availability of communications among the entities at the same level and between the entities at the lower level and the Master or superior level.

1.2.2 *Temporal Decompositions:*

In contrast to the physical disaggregation described above, temporal hierarchical decompositions are employed to simplify real-time, closed-loop planning problems [9, 10]. For these cases, the decomposition is characterized by higher levels that create plans with the greatest temporal scope (longest planning horizon) but with the least detail. At lower levels, the planning horizon becomes shorter (nearer term), but the level of detail of planned activities increases. The less detailed plans at the higher levels coordinate or guide the generation of solutions generated at the lower levels.

Indeed, planning actions over extended periods of time at a high level of detail is typically both futile and impractical: *futile* because detailed actions planned on the basis of a specific prediction of the future may become obsolete well before they are to be executed due to an inability to accurately predict the future, and *impractical* because the computational resources required to develop detailed plans over extended periods of time may be prohibitive either in cost or availability or both. The relationship between the levels of the hierarchy and the planning horizon and level of plan detail is shown in Figure 1.6.

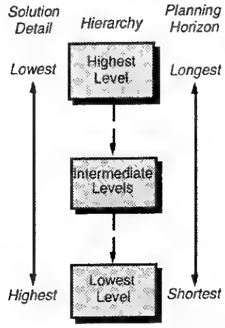


Figure 1.6 Characteristics of Solutions at Various Levels of the Hierarchy

1.3 Command and Control

Figure 1.7 illustrates an interpretation of the functional decomposition from the perspective of Command and Control (C^2). From this interpretation, the normal operational loop entailing *plan execution - system-to-be-controlled - monitoring* can be viewed as the control component of C^2. The outer loop whose functions are initiated by exception includes the *diagnosis - plan generation - plan selection* functions and can be viewed as the command component of C^2.

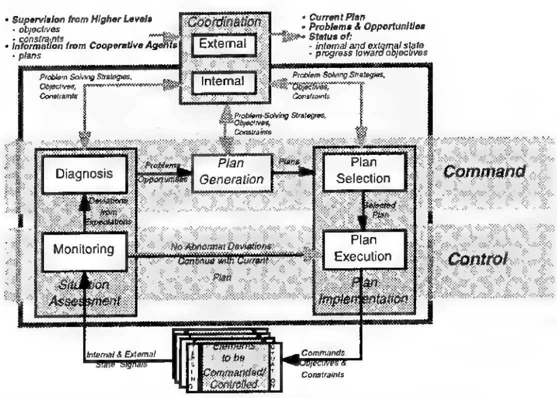


Figure 1.7 Command and Control Processes

1.4 Problem Solving

Thus far, the discussion of the functional decomposition of a real-time process has centered around applications of planning and decision-making. In addition, one can map a general closed-loop problem-solving scenario onto that decomposition. In this case, *plans* are expressed as *solutions* and the *system-to-be-controlled* is replaced by a *work object* (e.g., a machine tool fabricating a part, a robotic manipulator assembling a large space structure, etc.). Figure 1.8 depicts the functional analysis in this context of problem solving. The functions of each of the individual elements of the decomposition parallel those described earlier.

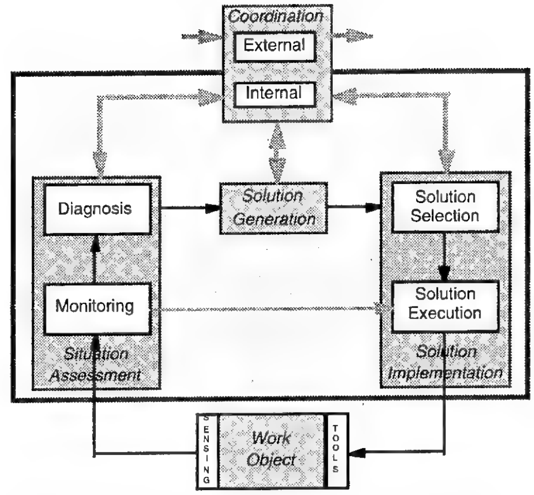


Figure 1.8 Functional Decomposition: General Problem Solving

1.5 Cooperative Planning Agents

In many situations it may be either required or desirable for a team to create and execute the solution to a real-time problem. This implies that function allocation is constrained to reflect a team solution. Although the discussion presented here applies equally well to a person-person team, of particular interest is the person-machine team wherein a person and a computer act cooperatively in solving a problem.

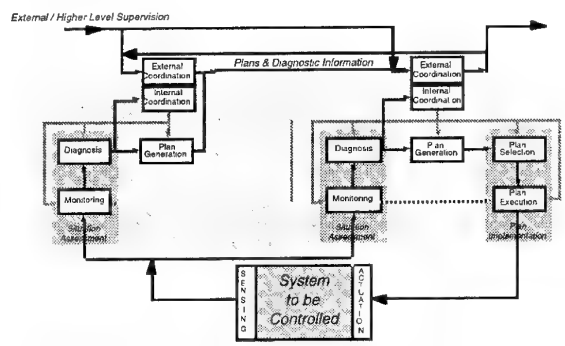


Figure 1.9 Cooperative Planning Agents

Supervisory control of an unmanned vehicle falls within this class of problems. Figure 1.9 illustrates a modification to the functional analysis presented earlier that accommodates this cooperative approach. In that figure, we have assumed that only the Situation Assessment and Plan Generation functions are executed cooperatively. The entity on the left in that figure, makes an independent assessment of the situation and, if necessary, generates independently a plan or set of plans. Both the assessment and generated plans are shared with the dominant agent (shown on the right) who is responsible for selecting and implementing a single plan. Note that for the case of person-machine cooperation, either the person or the machine could be the dominant agent, and, as discussed earlier, the

dominant role may reverse depending on the situation.

1.6 Reference Following Control

The final illustration of the applicability of the generic functional analysis is for a classical reference following control problem. This class of problems represents an early example of the application of automation to a real-time, closed-loop process. Figure 1.10 shows the mapping of each of the individual decision-making functions identified earlier onto the control problem: i.e., Monitoring, Diagnosis, Plan Generation, Plan Selection and Plan Execution. This mapping reinforces the view that the problems addressed here are a natural abstraction of those traditionally addressed in aircraft guidance, navigation and control. The ability to realize these abstractions in a machine implementation has been enabled by advances in computational hardware and algorithmic and knowledge-based systems approaches to developing solutions to the associated real-time problems that have been identified here.

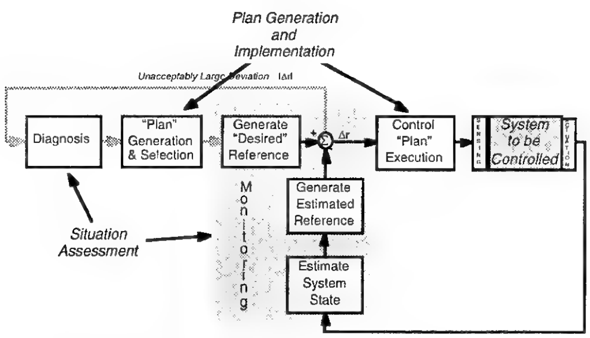


Figure 1.10 Functional Decomposition: A Control System Perspective

2 HIERARCHICAL DECOMPOSITION FOR MISSION PLANNING AND OPERATIONS MANAGEMENT FOR A MIXED FLEET OF MANNED AIRCRAFT AND UTAS

A hierarchical functional decomposition is described for mission planning and operations management for a mixed fleet of manned aircraft and Unmanned Tactical Aircraft (UTA). The capabilities that are envisioned to be embodied in the UTAs range from reconnaissance and battle damage assessment currently being performed by UAVs (unmanned air vehicles) as well as targeting and weapons delivery. The aeroperformance of UTAs is expected to be of the class of current tactical jet aircraft and could include low-observable stealth technology as well.

2.1 Hierarchical Command and Control

A hierarchy of command and control will be required to support the UTA force structure described herein. At the lowest levels will be the UTAs themselves. Directly above the UTAs are the individual pilots/mission controllers who have communication line-of-sight to the UTAs, with each pilot controlling / supervising / coordinating 5 to 10 UTAs. Above the individual pilots is the Air Component Authority and his staff who coordinate among regions and are responsible for airspace control.

Figure 2.1 illustrates several levels of a hierarchy comprising a command level air-based or ground-based air operations management function required to coordinate a group of airborne human pilots (air battle management level), each, in turn, coordinating the activities of a group of UTAs. Note that this architecture restricts neither the mix of human operated aircraft at the middle level nor the mix of UTAs under their supervision.

Mission and trajectory plans are developed at the air vehicle levels within the hierarchy (both the human piloted air battle manager and the UTAs under his control) to optimize an established objective function (e.g., minimize fuel, minimize time or maximize a mission-specific measure of accomplishment) subject to specified constraints (e.g., allocations on mission timelines, fuel, flight safety, etc.). A further hierarchical decomposition of the mission planning problem is envisioned, wherein skeletal plans of the entire mission of a given air vehicle is constructed at the highest level, the *Mission level*. The skeletal mission level plan must be generated at a sufficient level of detail to insure that onboard resources are sufficient to achieve the planned objectives and that timeline and survivability constraints are honored. At intermediate levels, the *Route/Activity levels*, near-term actions that are consistent with the mission level plan are planned in greater detail. Finally, at the lowest level of the hierarchy, the *Flight Safety level*, very near term commands are generated for sensor and control systems in a manner that ensures flight safety.

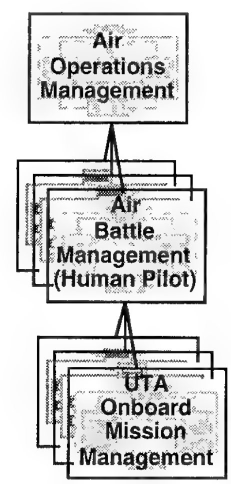


Figure 2.1 Air Operations Management Hierarchy

Figure 2.2 shows two levels of such a decomposition. The upper level creates mission plans spanning the entire mission and the lower level fills in the details of trajectory and payload activities that are required in the near term in pursuit of the mission plan. In

the figure, the farther term plan-generating-entities at the lower level are shown in gray to emphasize that although **in theory** the lower level would generate the detailed trajectory/activity plans for the entire mission, **in practice** only the near term plans are, in fact, produced.

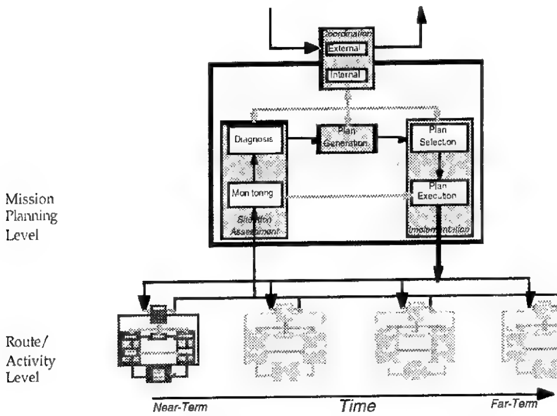


Figure 2.2 Hierarchical View: Upper Level: **Mission Planner,** Lower Level: **Route/Activity Planners**

2.2 Varying Degrees of Autonomy

In order to blend productively with manned aircraft, several levels of autonomous and semi-autonomous control are needed. They would allow UTAs to operate autonomously, but with human intervention on an "exception basis," when human decisions are needed. Thus, in order to successfully orchestrate the activities of an internetted flight of UTAs, the mission controller may occasionally be required to intervene in the functionality of individual UTAs at varying lower levels of control (i.e., lower than the command level at which the human normally interacts).

The implications of this are twofold. First, under normal situations the UTA must be able to autonomously plan the details of its mission in a manner that is consistent with the high level commands/intentions/objectives imparted to it from the pilot mission controller. Second, the UTA must be able to determine when the situation has evolved to a state that is beyond its ability to successfully cope with it on its own (i.e., autonomously). In those cases, it must be able to communicate those difficulties to the human and mechanisms must be provided to allow the human to intervene at various levels within the UTAs own onboard hierarchy of control.

Assume that there are three onboard levels: mission, route/activity and aircraft control (see Figure 3.2). The overall system must be designed to allow the human to intervene at all levels. The intervention could be initiated from an onboard request by the UTA when it "understands" that it is incapable of coping with an undiagnosable situation or externally by the human when, by monitoring of a UTAs current actions or planned activities, he decides that a change must be made. For a self-initiated intervention, the UTAs onboard software must be designed to detect these circumstances, communicate a description to the human and expect an intervention. If none evolves, this must also be determined and a temporary abort of the current mission activity initiated. Interventions by the human at the highest level are most consistent with the "command" role of the human, but interventions at the lowest level may be required when, for instance, the human determines that more information should be gathered regarding a specific target from a specific aspect or if he desires to override the normal UTA delivery of munitions.

3 SOFTWARE FRAMEWORK FOR AUTONOMOUS VEHICLE ONBOARD PLANNING

A generalized software framework for implementing some of the elements of the hierarchical systems described herein has been developed for onboard mission planning for autonomous vehicles under internal funding at the Draper Laboratory over the past several years [11, 12]. Having realized that there is a domain independent component of the software that is required for the general management of the processes of monitoring, diagnosis, planning and plan execution which is both mission and vehicle independent, Draper undertook an effort to develop a reusable implementation of that component. A problem-specific planning system (i.e., a vehicle and mission-scenario specific system) is created from the framework when the framework's knowledge and data bases are augmented with information about the vehicle and its environment. The framework alone contains knowledge about real-time planning system processes. The framework has been employed for several autonomous vehicle applications including ground and underwater vehicles.

This separation of the knowledge required for coordinating and implementing the generic planning system processes from the knowledge about the specific controlled vehicle and its environment allows the planning framework to be rapidly customized and integrated with the onboard software of the controlled system. The framework supports use of an arbitrary number of hierarchical levels of abstraction, performing the planning, execution, execution monitoring, and replanning functions at each level (see Figure 3.1 for a two level, two controller system).

The framework maintains long term abstract plans at its top level to ensure satisfactory overall mission performance and overall consistency in resource allocation; it concurrently maintains shorter term detailed plans at all levels below, each level a more detailed consideration of the activities of the level above. The plans at the bottom level are sent to the control system of the plant for execution, and the progress toward completion of the activities in the plans at all levels is monitored by the framework and assimilated into the projections of plan

performance that are **maintained at each level**. The framework monitors these projections of potential plan performance for indications of potential to improve the plan at any level. Should adequate potential for improvement exist, the planning system replans accordingly.

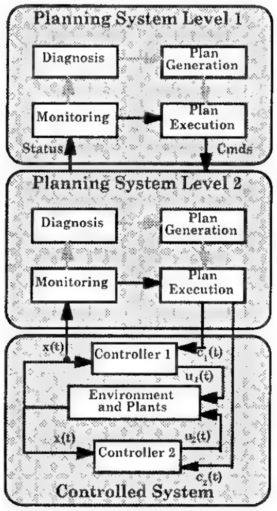


Figure 3.1: Two Level Hierarchy

Mechanisms have been built into the framework for controlling the time horizon of the planning problem to be solved at each level and for selecting the time to spend generating a plan; however, the heuristics that have been designed to date to adjust these values are primitive. In addition, it is often desirable to adjust the "greediness" of the planning process; a greedy algorithm can find a reasonably good answer to a large planning problem in a relatively short amount of time, but to find a better solution one must use a more thorough, albeit slower to completion, algorithm.

The decisions that lead to effective replanning upon recognition of a problem or unexpected opportunity include: (1) the amount of time that should be spent planning and (2) the magnitude or extent of the planning problem to attack. The magnitude of the planning problem generally increases with the problem's size (level of detail and time horizon) and decreases with the planning algorithm's greediness. In cases of impending threats to system safety or integrity, the choice of a short amount of time to solve a small problem can serve to buy time for a successive phase of planning wherein a longer time (the "bought" time) is used to solve a planning problem of more significant magnitude (i.e., horizon), whereas in cases of smooth operation and little uncertainty it may be more useful to initially allocate a large amount of planning time to work out details of the plan far into the future. The capability to tailor the type of planning undertaken to the situation at hand allows the system to better solve the problem of maximizing achieved mission value.

Ongoing efforts directed toward improvement of the planning framework consist of two thrusts: one is to improve the algorithms and heuristics employed by the planning framework to diagnose problems and define the type of planning appropriate for the situation at hand; the other is to provide the planning system with controls to adjust the tradeoff between the thoroughness and speed of plan generation. To these ends, the following tasks are underway:

(1) Develop mechanisms to control use of low level plan results during upper level planning.

(2) Enhance implementation of the framework's ability to use contingency plans.

(3) Develop and evaluate heuristics to determine the amount of time to allocate to planning.

(4) Enhance the overall capability of the framework in planning to plan (metaplanning).

(5) Develop and evaluate heuristics to determine the desired time horizon, level of detail and planning thoroughness of the plan generation process.

3.1 Use of low level plan results during upper level planning

The knowledge base associated with the planning framework is broken down hierarchically into levels, each of which corresponds to the level of abstraction of a plan to be maintained by the framework. Each level of the knowledge base consists of two principal parts: *activity models* which describe the effect that performing an activity has on the state of the system, and *activity planners* which describe how an activity at the level is to be decomposed into goals for the next inferior level down (the level below the lowest level is the control system).

Presently, the framework's planning algorithms rely on the activity models to determine the expected outcome of potential plans generated during the planning process. The models, data, and information employed at the higher levels of the framework are abstracted from the detailed information used at lower levels. As a practical matter, however, it is sometimes computationally prohibitive or simply not feasible to abstract all the information that the upper levels might need for planning from the lower level information. Under this proposed effort the planning framework would be made capable of invoking the planning process at the lower levels to determine more precisely the results that implementing a potential high level plan would produce. As a matter of efficiency, the results of invoking the lower levels would be stored so that they could be used again, when appropriate.

The modifications are required to provide a *planning thoroughness* control for the planning system to adjust the conditions under which the planning algorithm would invoke the lower level planners rather than relying on the activity models. Adjustment of this control would allow the planning system to specify that planning be performed in a manner ranging from being based solely on the

activity models for a fast answer (as it is now) to being based solely on the results of lower level planners in an iterative search of the type employed in mathematically formal multi-level optimization solution methodologies.

3.2 Contingency Plans

Currently, the framework provides the mechanisms to represent contingency plans and to calculate expectations of utility and resource use for a plan with embedded contingency plans. A modification is required to allow the selection and planning of contingent scenarios, and a mechanism is required to determine when circumstances warrant making contingency plans.

3.3 Heuristics to Determine Allocation of Planning Time

The heuristic that determines the amount of time to allocate to plan generation is being improved. Essentially, the heuristic compares an estimate of the value (expected utility) that would be achieved by a plan created in the given amount of planning time to an estimate of the utility to the mission irretrievably lost by following the existing plan for that same amount of time. This, of course, must be addressed for every level of the planning system hierarchy.

3.4 Planning for Time to Plan

Planning for time to plan requires that the plan generation algorithm estimate the improved value likely to be achieved by replanning during execution of future parts of the plan. The planning system must look one planning cycle (across the hierarchy) ahead while planning to plan.

3.5 Determining the Nature of the Planning Process

A heuristic is required to determine, given an amount of time to plan, the settings of the planning thoroughness parameter and the sensitivity / time horizon threshold that is employed in triggering contingency planning.

4 REFERENCES

[1] Antsaklis, P.J. and Passino, K.M., (eds.), *An Introduction to Intelligent and Autonomous Control*, Kluwer Academic Publishers, Boston/Dordrecht/London, 1992.

[2] Sacerdoti, E., *A Structure for Plans and Behavior*, Elsevier, North-Holland, New York, 1977.

[3] *Combat Automation for Airborne Weapon Systems: Man/Machine Interface Trends and Technologies*, AGARD Conference Proceedings of the Joint Flight Mechanics Panel and Guidance and Control Panel Symposium, Edinburgh, Scotland, October 1992, AGARD-CP-520.

[4] Rouse, W.B., Geddes, N.D. and Curry, R.E., "An Architecture for Intelligent Interfaces: Outline of an Approach to supporting Operators of Complex Systems," *Human Computer Interaction*, Vol, 3, 87-122, 1988.

[5] Lasdon, L.S., *Optimization Theory for Large Systems*, Macmillan, New York, 1970.

[6] Haimes, Y,Y et al., *Hierarchical Multiobjective Analysis of Large-Scale Systems*, Hemisphere Publishing Corporation, New York, 1990.

[7] Mahmoud, M.S, Hassan, M.F. and Darwish, M, G., *Large-Scale Control Systems: Theories and Techniques*, Marcel Dekker, New York, 1985.

[8] Drouin, M., Abou-Kandil, H., and Mariton, M., *Control of Complex Systems*, Applied Information Technology Series, Plenum Press, NY, NY 1991.

[9] Albus, J. S., Lumia, R., and McCain, H., "Hierarchical Control of Intelligent Machines Applied to Space Station Telerobots," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 24, No. 5, September, 1988.

[10] Adams, M.B., and Beaton, R. M., "Planning and Planning Management for Autonomous and Semi-Autonomous Vehicles," *Proceedings of the AGARD Guidance and Control Panel 51st Symposium*, Knowledge Based System Applications for Guidance and Control, Madrid, Spain, Sept. 18-21, 1990, AGARD-CP-474.

[11] Hall, William and Adams, Milton , "AUV Software Taxonomy", Proceedings of the 1992 Symposium on Autonomous Underwater Vehicle Technology, Washington, DC, 1992.

[12] Hall, William , "Resource-Coordinated Hierarchical Planning for Real-Time Autonomous Systems," SM Thesis, Massachusetts Institute of Technology, 1992.

Distributed, Collaborative, Knowledge Based Air Campaign Planning

Mark T. Maybury
Bedford Artificial Intelligence Center
Advanced Information Systems Technology Department
The MITRE Corporation
Mail Stop K331. 202 Burlington Road
Bedford, MA 01730-1420 USA
Tel: (617) 271-7230
Fax: (617) 271-2352
E-mail: *maybury@mitre.org*

SUMMARY

This paper addresses existing functional needs and current technical opportunities for intelligent automation to support air campaign and theater level planning. In the context of a changing political, military, and acquisition environment, we describe several advanced automation activities that address key shortfalls in situation assessment, force planning, and legacy systems integration. First we describe a joint Air Force Electronic Systems Center (ESC)/MITRE Corporation effort to deal with the "legacy" problem of integrating intelligence and mission planning systems using a common object request broker architecture to enhance intelligence/operations interactions and support evolvable systems in the field. We then describe results from a joint Advanced Projects Research Agency (ARPA) and Rome Laboratory (RL) initiative aimed at developing the next generation of distributed, collaborative force deployment and force employment planning technology. We then describe another ESC/MITRE effort to develop tools for multisource intelligence integration to support knowledge based, multisensor data fusion and enemy behavior recognition for enhanced situation assessment. Given this context, we then illustrate an integrated vision of a distributed collaborative, knowledge based crisis action planning system, where both machine and human knowledge are utilized synergistically to enhance overall system performance. We summarize lessons learned from these efforts and discuss an evolutionary acquisition process to move the above ideas toward operational realization while minimizing technology transition risk. The article concludes with recommendations for moving forward.

1. INTRODUCTION

United States national security policy states a requirement "in concert with regional allies, to win two nearly simultaneous major regional conflicts" [1]. Supporting this objective requires a revolutionary approach to joint and coalition doctrine as well as significant advances in supporting Command Control Communications and Intelligence (C^4I) infrastructure. Of critical importance to sustain the initiative in warfare is our ability to stay inside the enemy's planning cycle time. While we are increasingly able to work in concert with our allies at a political level to control the proliferation of weapons of mass destruction and to promote stability and democracy, in battle we remain limited in our ability to share information and collaborate using an electronic information infrastructure.

This article highlights recent technology developments and novel acquisition strategies that attempt to address this need. It emphasizes future distributed mission planning and execution infrastructure which can be used to collaboratively plan air campaigns. While the primary focus here is on joint US systems and in particular force deployment and employment, lessons learned may be transferable to coalition activities.

2. PROBLEM

Supporting the US national security strategy of enlarging the global village of free market economies, American troops operated in nearly every country in the world in 1994. For example, in the Air Force:

> We delivered 75,000 tons of relief supplies to Bosnia, 15,000 tons to Rwanda and Zaire; supported major deployments to Haiti and Kuwait; and conducted hundreds of operations in such far-ranging places as Yemen and Johnston Atoll ... We've flown nearly 10,000 sorties in Bosnia. In the Gulf we've launched three times the missions of Desert Storm. Within 10 days of Iraq's provocation this Fall, 160 combat aircraft joined the 140 already deploy there, and we had flown 1,000 sorties ... We've exercised with 50 nations since last December. [2].

These activities underscore the multifaceted nature of modern military operations, spanning tradition roles to defend against, deter, damage/disable, or destroy enemy threat as well as to engage in combat operations other than war, including relief missions and non-combatant evacuation operations.

Desert Storm illustrated the effective application of coalition air power, stealth technology, precision-guided munitions, and C^4I to achieve decisive victory. Despite this success, lessons learned suggest a clear need for a more integrated view of the battlefield to better perform situation assessment, more timely and accurate force deployment and employment, and a more efficient information systems infrastructure to enable rapid plug-and-play of capabilities.

Finally, guidance from Secretary of Defense William Perry emphasizes the use of standards to promote interoperability, Commercial Off the Shelf (COTS)

Paper presented at the Mission Systems Panel of AGARD and the Consultant and Exchange Program of AGARD held in Madrid, Spain, on 6-7 November 1995; Châtillon, France, 9-10 November 1995; and NASA AMES Research Centre, California, USA, 16-17 November 1995 and published in LS-200.

solutions to reduce costs, and joint infrastructure and architecture. Important capabilities are now fielded such as the commercially-based Joint Worldwide Intelligence Communications System (JWICS) which provides interservice video, voice, and data connectivity, and the Joint Deployable Intelligence Support System (JDISS), a DODIIS core project which provides the JTF commander with a common UNIX-workstation suite (e.g., e-mail, file transfer, remote access, imagery). While the move toward COTS provides important functional and economic advantages, it is not without risk. In addition to marketplace volatility, experience suggests that effective COTS integration requires detailed knowledge of and access to internal and potentially proprietary source code. Furthermore, there remain functional gaps between desired concepts of operations and government and/or COTS systems as well as serious interoperability problems with existing and projected operational support systems. Crucial to a successful information infrastructure is a well articulated target architecture as we move toward a Global Command and Control System (GCCS).

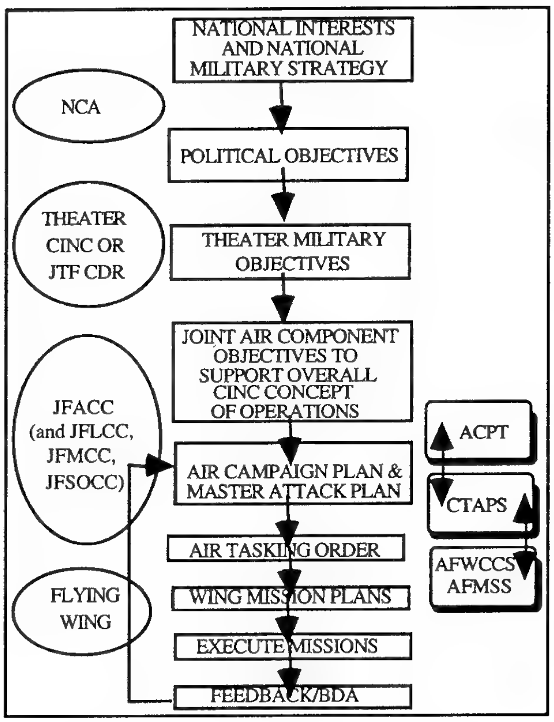


Figure 1. Battle Planning Process

In the remainder of this article we first outline the promise of distributed object computing technology and how this is being exploited to improve future theater level mission planning systems. We then point to innovative joint developments for new distributed, collaborative, knowledge based planning aids at the air campaign level. As Figure 1 illustrates, we focus on current theater level automation systems first, in particular the Air Force's Contingency Theater Automated Planning System (CTAPS). We subsequently turn our attention to future air campaign level automation such as the Air Campaign Planning Tool (ACPT) which produces an overall air campaign plan and a daily Master Attack Plans (MAPs), a potential future input to CTAPs. We do not directly address mission level automation systems such as the Air Force World Wide Command and Control System (AFWWCCS), through which Air Force Wings receive Air Tasking Orders (ATOs) from CTAPS, nor the Air Force Mission Support System (AFMSS), which is used by air crews for tactical mission planning.

3. CTAPS

For many reasons, including changing threats, doctrine, concept of operations, and resources, many large systems are procured to function independently only to discover a future need to interoperate. Figure 1 illustrates CTAPS, a complex, system of systems indicative of the current complexity of theater level infrastructure. For Air Force theater-level battle management, CTAPs is at the heart of the cycle of situation monitoring, diagnosis, plan generation, plan selection, and plan execution, as articulated by NATO AGARD Working Group 11 [3]. CTAPS contains approximately two and one half millions lines of source code encompassing multiple mission functions (from situation assessment to weaponeering to battle planning), software applications (e.g., heterogeneous databases, human computer interfaces), and programming languages (e.g., C, C++, Ada, SQL, Pro C).

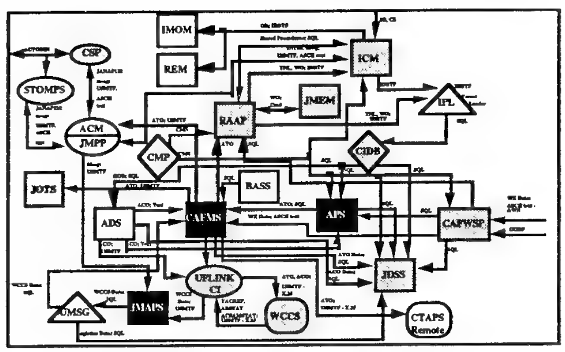


Figure 2. CTAPS Architecture

ESC and The MITRE Corporation in a Mission Oriented Investigation and Experimentation (MOIE) project [4, 5] are examining the integration of legacy CTAPS systems via coarse encapsulation of application objects using the Common Object Request Broker Architecture (CORBA), described below. The motivation is not only to consolidate existing systems in order to reduce cost, increase information consistency, and improve responsiveness. It is also to establish a computational framework which will enable rapid migration to new requirements and systems, including some of the advanced distributed campaign planning tools outlined in the next section.

As highlighted in Figure 2 , MITRE has experimented with the ease and utility of the CORBA integration of three CTAPS subsystems: the Computer Assisted Force Management System (CAFMS), Advanced Planning System (APS), and Joint Message Analysis and Preparation System (JMAPS), together which constitute a half million lines of source code. Currently these

applications are coarsely encapsulated using IONA Technologies' Orbix environment as individual application objects, that is there exists a CAFMS object, an APS object, a JMAPS object. Future work will address integrating individual functions and data within these systems, for example, supporting interaction with a mission plan object or an enemy threat object.

3.1 CORBA and CTAPS

The Object Management Group (OMG), formed in 1989, is a consortia of over 500 member companies including the major software system vendors (e.g., Apple, IBM, Digital, Hewlett-Packard, Microsoft, and SunSoft, on whose operating system CTAPS runs) and large end-user organizations which aim to support interoperable software components in heterogeneous environments via the development of standard interfaces and supporting infrastructure. Their resultant reference architecture is based on objects which have associated operations. It further distinguishes object services such as the general management of objects (e.g., their creation, deletion, naming, copying, querying, modification), from common facilities (e.g., object browsers, user interface components, mail, print spoolers, spelling checkers, help facilities) which may be reused in multiple applications, from application objects, which would be custom to a particular domain.

An Object Request Broker (ORB) acts as a communications infrastructure to route messages between objects in a manner independent of the language, platform, and networking protocol local to any object. An Interface Definition Language (IDL) is used by object developers to define the language-independent interface to an object type and an Interface Repository acts as a database of object interface definitions as well as data types, constants, and exceptions. A Dynamic Invocation Interface enables a client, at run-time, to invoke an arbitrary operation on an arbitrary type of object. Inter-ORB protocols were adopted in December 1994, most importantly, the Internet Inter-ORB Protocol for interoperation among ORBs via TCP/IP. Forthcoming extensions will include mappings from IDL to additional languages beyond C, C++, and Smalltalk (e.g., Ada9X, COBOL, LISP, Objective-C).

Figure 3 outlines the Object Request Broker architecture as applied to CTAPS. Object Services are similar to those found in the general CORBA model, however, facilities include both general items (e.g., system administration, e-mail, talk) as well as ones particular to military operations (e.g., message processing, alerting, mapping). In contrast, application objects are unique to theater level mission planning (e.g., theater intelligence, air tasking order planning, weaponeering).

MITRE wrote IDL interfaces and implemented CORBA front-ends for JMAPS, CAFMS, and APS. For example, via ORB invocations, the APS application can be invoked and exited, with or without a map, and the APS data export application can be invoked. Once an Air Tasking Order (ATO) is prepared in CAFMS, the JMAPS ATO Check function can be invoked via the ORB, which passes the ATO message as input to JMAPS and receives an error report as output, also via the ORB. These CORBA front-ends represent encapsulations of applications with command-line (APS and CAFMS) and remote procedure call programming interfaces.

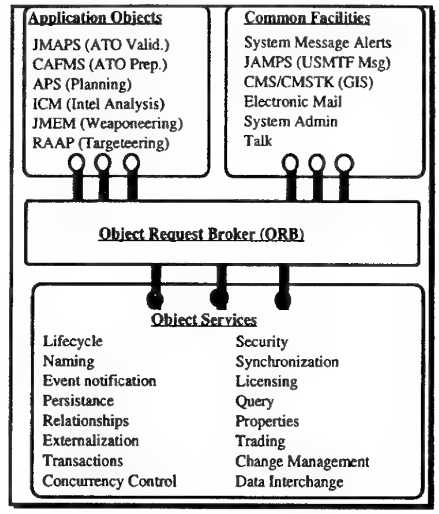


Figure 3. CTAPS ORB Architecture

The ORB also can act as an intermediary not only to applications but also to data. For example, MITRE wrote IDL interfaces and developed a CORBA front-end for relational database management systems. This was specialized to access Oracle. MITRE then developed a front-end based on Mosaic software from the National Center for Supercomputing Applications. This front end supports direct access through an ORB interface to APS and CAFMS databases.

3.2 Lessons Learned

Wrapping existing legacy systems by defining and implementing CORBA interfaces provides a powerful method for systems migration. Coarse encapsulation of legacy systems using CORBA does not require access to the source legacy code, provided sufficient knowledge of high level interfaces. Indeed, it is application architecture knowledge (components, their functions, characteristics, operating assumptions, and interactions) that required the most amount of resources in the above experiment. In fact, the source code to develop the Orbix IDL definitions, servers, human computer interface and utility functions for APS, CAFMS, and JMAPS totaled only 2,617 lines of code (contrast this with the half million lines of code represented by those applications). With object-oriented access to legacy system data and functionality, we have the possibility of moving up the planning systems support hierarchy shown in Figure 1 toward distributed, collaborative planning tools as shown in Figure 1. We turn to these next.

4. AIR CAMPAIGN PLANNING

Figure 4 provides a more detailed view of the levels, inputs, decisions, and activities from campaign planning to execution [3]. Just as the wing and flight level operations require detailed intelligence about terrain, weapons, threats and weather to plan an effective mission, theater and

campaign level planners require tools to support situation assessment, course of action development, evaluation, and selection. In this section we will describe several systems that support deployment and employment planning.

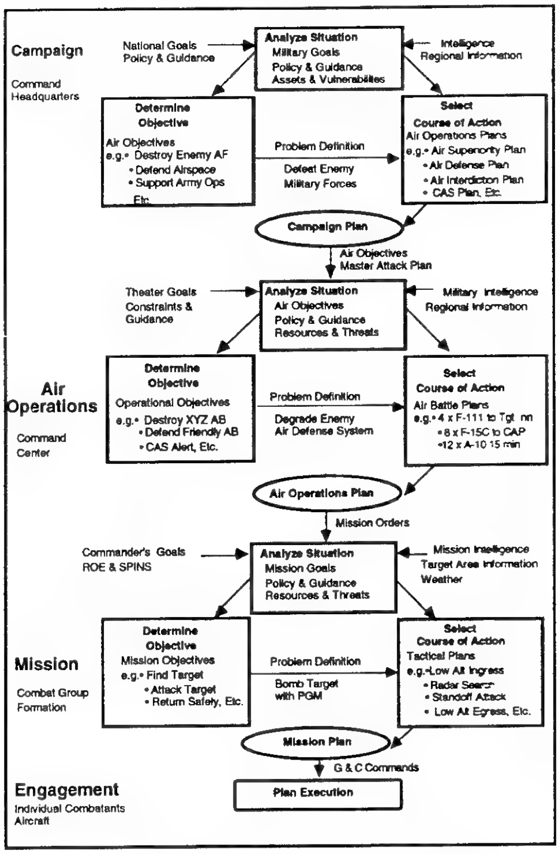


Figure 4. Air Campaign Planning Process [3]

4.1 ARPA/RL Planning Initiative

The joint Advanced Research Program Agency (ARPA) and Rome Laboratory (RL) Knowledge based Planning and Scheduling Initiative (ARPI) is aimed at developing the next generation of distributed, collaborative planning tools [6]. ARPI takes a multi-tiered approach to development via problem focused basic research which flows into Integrated Feasibility Demonstrations (IFD) which then flow into Advanced Capability Technology Demonstrations (ACTDs) which in some instances are fieldable capabilities. Formal "exit" criteria and "final exams" serve as functional evaluations at each step in the process.

IFD-1 consisted of the Dynamic Analysis and Replanning Tool (DART), which was used by transportation planners to manipulate Time Phased Force and Deployment Data (TPFDD). This included the TPFDD Editor (TPEDIT), a temporal constraint-based tool used to construct and edit the elements and temporal aspects of the TPFDD. TPFDDs consist of many unit line number (ULN) records. Figure 5 shows a sample ULN for a US Army air defense artillery battery of Patriot missiles originating from HCRL on C000 (estimated), embarking at NKAK on C010 (estimated), with a destination of JEAH between C011 and C015 (estimated). Built by a team that integrated end users and technologists, operational folks found DART construction, analysis, and interface showing transportation phasing and feasibility extremely useful. It was claimed a major success in its use for transportation planning during Desert Storm [6].

```
((ULN "U-0AADA ")
(PROVORG "7") (SERVICE "A")
(UTC "1HM77") (ULC "BTY")
(DESC "ADA BTRY,PATRIOT (MISSLES)")
(FIC "8") (PIC " ") (BULK "0000000")
(OVER "0000450") (OUT "0000000")
(NONAIR "0000000") (ORIGIN "HCRL")
(RLD "C002") (POE "NKAK")
(ALD "C010") (EDD "C000")
(POD "JEAH") (EAD "C011")
(LAD "C015") (PRIORITY "002")
(DEST "JEAH") (RDD "C015")
(SEQNBR "00000") (CEI " "))
```

Figure 5. Sample ULN record from TPFDD database

IFD-2 developed a complex knowledge based planning tool, the SIPE-II Operational Crisis Action Planner (SOCAP). SOCAP consisted of a set of planning operators that specified a taxonomy of military actions (e.g., deploy a unit, perform a mission, allocate a route) each of which achieve particular goals. Each action had associated resource, temporal, and activity constraints. Given a high-level operational objectives, SOCAP could generate an hierarchical plan of air campaign actions.

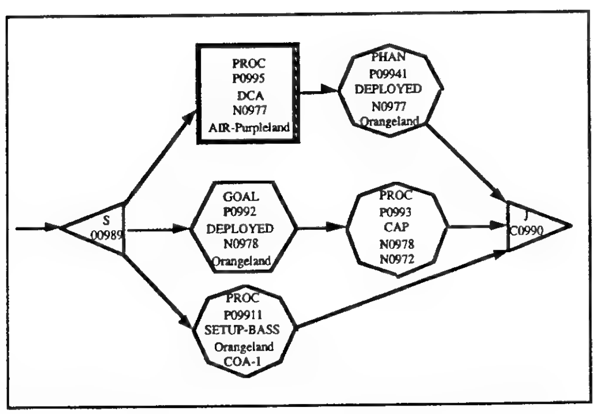


Figure 6. Segment of SOCAP Plan

Figure 6 shows a portion of a partially planned SOCAP course of action to defend territorial integrity. In Figure 6, triangles indicate the start or completion of parallel actions, squares and hexagons indicate processes and goals which require further refinement, and octagons indicate processes (with associated actions, resources, and results). Thus, the partial plan in Figure 6 consists of three parallel steps (from bottom to top):

1. Setting up a base at Orangeland.
2. Deploying tactical fighter forces N0978 (a yet unspecified unit) at Orangeland, followed by a Combat Air Patrol (CAP) performed by N0978 and N0972
3. Deploying tactical fighter forces N0977 for Defensive Counter Air (DCA) in Purpleland, followed by a CAP performed by N0977.

This plan portion could be further refined and/or replanned to achieve higher level goals.

Detailed courses of action are generated by SOCAP by reasoning about goals and detailed action specifications. For example, the tactical airlift operator used to plan moving *force1* from *airfield1* to *airfield2* is:

```
OPERATOR:         move-by-tairlift
 ARGUMENTS:       army1-with-size<10000,
                  airfield1-with-runway>500,
                  airfield2-with-runway>500,
                  tairlift1, tfighter1, air-loc1;
 PRECONDITION:    (route-aloc airfield1 airfield2 air-loc1);
 PURPOSE:         (moved army1 airfield1 airfield2);
 PLOT:
   GOAL:          (located tairlift1 airfield1);
   PROCESS
   ACTION:        move-tairlift;
   ARGUMENTS:     army1, airfield1, airfield2, tairlift1;
   RESOURCES:     tairlift1;
   EFFECTS:       (moved army1 airfield1 airfield2);
 END PLOT
END OPERATOR
```

This detailed plan specifies the context of a successful tactical airlift. For example, the preconditions for successful application of this plan dictate that force1 must be smaller than 10000 tons, *airfield1* and *airfield2* must have runways longer than 500 feet, and there must be an air corridor between *airfield1* and *airfield2*. The purpose of the act is to move *army1* from *airfield1* to *airfield2*. SOCAP demonstrated the feasibility of deliberative planning, although operational users had difficulty understanding PERT-like views of hierarchical plans (as in Figure 6) as opposed to GANTT-chart like views. A more complete library of plan operators needs to be constructed to deal with a variety of operational courses of action.

4.2 TARGET and ForMAT

Theater Analysis, Replanning and Graphical Execution Toolbox (TARGET) was developed and demonstrated during IFD-3. TARGET as well as other ARPI technology was demonstrated daily at the Joint Warrior Interoperability Demonstration (JWID '94), a Joint Staff sponsored annual forum focused on C^4I concepts, technologies, and systems [7]. The demonstration backbone was the Theater Analysis, Replanning and Graphical Execution Toolbox (TARGET) which provides such capabilities as shared plans, video, voice, maps, briefings and pointers. JWID '94 was used to demonstrate collaborative disaster relief planning in Hawaii at US CINCPAC and combat operations at USACOM and the Air Combat Command. TARGET includes a shared set of planning tools which enable users to jointly assess transportation feasibility, cost, casualties, and time associated with alternate courses of action (COA). In the relief scenario, a Combined JTF was simulated from NRaD in San Diego who interacted with surrogate members from Army Materiel Command and Defense Logistics Agency. Logistics, weather, and disaster anchor desks were provided on Oahu. Functionally, the planning tools enable crisis action members to rapidly produce Time Phased Force and Deployment Data (TPFDD), validate feasibility, visualize results on the Geographical Logistics Awareness Display (GLAD), obtain critical situation information from anchor desks, select a final course of action, and transmit this to the theater commander.

A crucial aspect to this process is selecting and supporting feasible courses of action (COAs). The Force Module Analysis and Management Tool (ForMAT) [8] was developed originally for deployment planners at CINCs to build the deployment plans for selected COAs. It is currently being explored for use as an adaptive Force Package editing tool and for supporting Service Components in force generation and selection. ForMAT is currently populated with 322 cases derived from 17 TPFDDs where each case contains elements from 47 possible attribute value pairs.

Using case based reasoning techniques developed for SMARTplan [9], the system is able to index, retrieve, support modification and visualize a database of TPFDDs based on high level specifications of force requirements (e.g., service=Army AND capability=anti-tank). Figure 7 illustrates a joint force created using ForMAT.

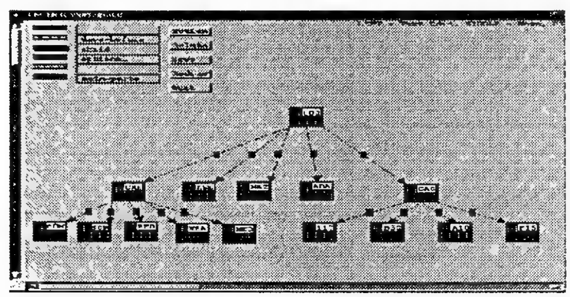


Figure 7. ForMAT Joint Force Structures

ForMAT represents a dramatic improvement both in the quality and speed of developing Operational Plans (OPLANs), which specify where and when forces involved in a mission are to be moved. This previously was a cumbersome process. Significantly, a "small" plan can involve specifying 10,000 distinct force requirements, a large one as many as 200,000, all of which much be scheduled. By matching desired requirements to similar previously stored solutions in the case base, prohibitive computations can be avoided. A user can query for an exact match to a force need (e.g., function=military-police AND service=air-force) and obtain a rank-ordered list or if there are few or no results, then the user can issue a "general" search which walks up a generalization hierarchy associated with search terms to broaden the search (e.g., function=security AND service=air-force). Instead of

querying the existing case base, the user can choose from "template force modules" to describe a generic force package (e.g., a small, medium, or large sized Marine Expeditionary Unit).

At JWID-94, ForMAT successfully received force requirements from TARGET and generated lists of satisfying forces. For example, after a mission planning session, TARGET would pass force requirements to ForMAT such as:

MISSION = DESERT-BLAST
THEATER = PACOM
GEOGRAPHICAL-LOCATION = KOREA
FUNCTION = MISSION-AIRCRAFT
SERVICE = AIR-FORCE
DEST-CC = WORLD
UIC = "WALOAA"

ForMAT would then retrieve a set of Force Modules prioritized by the degree to which they satisfied these individual and cumulative requirements. The Force Module functionality of ForMAT will be combined with the creation and editing functionality of TPEDIT and folded into the Global Command and Control System (GCCS). In addition to ensuring transportation feasibility, designing the campaign in the first place is a critical success factor to which we now turn.

4.3 Air Campaign Planning Tool

Current ARPI focus is on tools to support air campaign planning, in part a result of the success of the Air Campaign Planning Tool (ACPT) [10, 11], software developed for the USAF/XO and the "Checkmate" division therein by ARPA and the ISX Corporation. ACPT captures the process utilized during Desert Storm to help the Joint Force Air Component Commander (JFACC) and his staff rapidly build a high quality air campaign plan.

> Led by Lt. General Buster Glosson, a staff of USAF planners developed a strategic plan favoring the application of precision munitions against carefully selected "centers of gravity" to maximize the effect of limited force application, avoiding "mass-on-mass" application of force. [11]

Figure 8 illustrates a high level view of ACPT which indicates inputs, outputs, tools, and existing and envisioned interactions with external systems. ACPT helps the JFACC and his staff to:

- Perform situation assessment
- Specify campaign objectives
- Develop Courses of Action (COAs)
- Identify target Centers of Gravity (COG)
- Allocate resources
- Assess plan feasibility and effectiveness

For example, Figure 9 shows a screen dump in which an air campaign planner is specifying, refining, and satisfying an overall COA by selecting an action (in this case "attack"), an associated effect ("disrupt"), and COG.

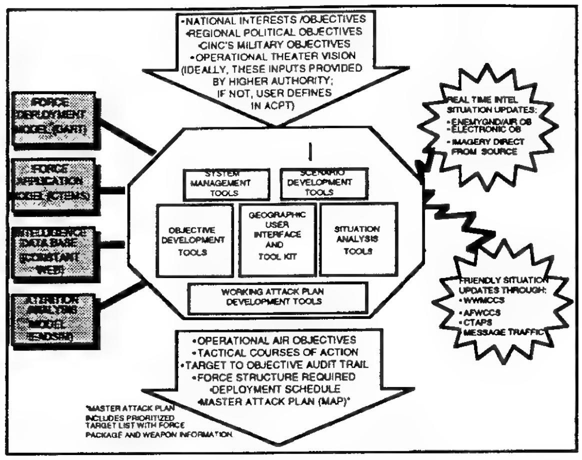


Figure 8. Air Campaign Planning Tool (ACPT) [11]

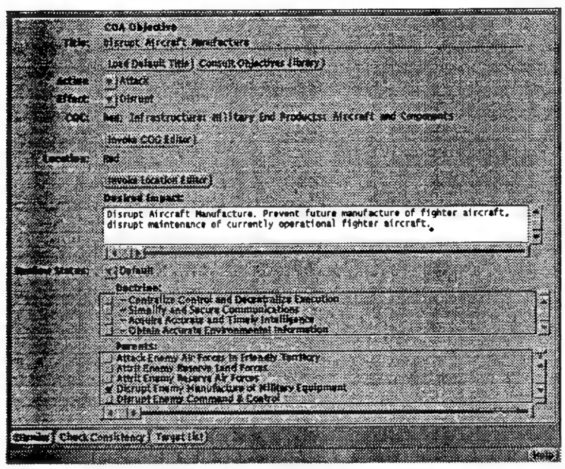


Figure 9. ACPT Assisting COA/COG Development[11]

Figure 10 shows the planner then selecting a target (e.g., all radar within 50 miles of a location) and exploring force requirements for particular COAs.

ACPT has the advantage that planners can build campaign plans during peacetime as well as during crisis, thus training with what they will fight. The high value application and access to a core set of experts were crucial to the success and continued daily use of ACPT.

Air campaign planning functions may be integrated into future versions of CTAPS for use at the Air Operations Center (AOC) level. Links between ACPT and CTAPS functions, as well as databases adequate to serve all applications, are challenges yet to be resolved.

4.4 Conclusion

JWID evaluations [7] of the above tools showed primary problems to be network capacity and reliability as opposed to functionality. Tools that are to effectively support the complex cognitive functions of analysis and planning need not only be intuitive, they also require detailed knowledge

of war fighting, a rich taxonomy of courses of action, and an ability to intelligently guide and support the planner. "Building in" knowledge acquisition to the process of campaign planning or force module retrieval/modification can ease the brittleness and cost of these tools, although capturing and representing situation/political context will remain difficult.

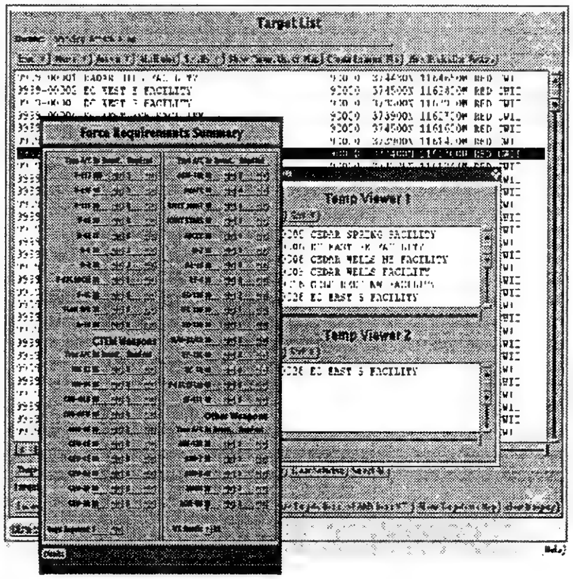


Figure 10. Target Selection and Force Requirements [11]

Finally, experience in ARPI underscores the power of user driven software development. Users are more likely to take a personal stake in the resulting systems they have influence over, and there is a higher likelihood of real increases in operational performance resulting in reduced costs and improved readiness given strong focus on actual problem areas.

5. INFORMATION FUSION

Knowledge of what you can do is only as good as knowledge of what you should do. Understanding the enemy or threat is crucial to plan selection. Often commanders have either too many minute pieces of information or too much general information from which crucial nuggets could be mined (e.g., open source). Moreover, as in the CTAPs system, multiple, separately developed systems can yield incomplete, inconsistent or even an incorrect view of the battlefield. In a separate MOIE effort called Multisource Intelligence Integration and Analysis (MSIIA) [12], The MITRE Corporation in concert with ESC has been investigating tools to assist the intelligence analyst in culling out a cohesive tactical picture of the battlefield.

The Joint Directors of Laboratories Data Fusion Subpanel [13] have developed a four-level generalized processing model that provides a common reference for discussing data fusion systems. The lowest level, Level 1, is represented by sensor-to-sensor correlation technologies and output products such as an estimate of an object's position and identity. At Level 2, logical processes use object information, order-of-battle, and environmental data to determine patterns and produce an assessment of the current hostile or friendly military situation. Products from Level 3 estimate the threat's capability and intent, and an emerging Level 4 addresses collection management. A number of sensor fusion systems are emerging in the intelligence area; however, they are limited in the number of sensors they process or their level of reasoning. A good example of this is the Extended Intelligence Support Terminal (X-IST) being developed for the Navy. X-IST correlates SIGINT and provides graphical representations of tracks on DMA raster maps. Because of the lack of vector map data, X-IST cannot reason about map features. X-IST has a video window and can manipulate softcopy imagery in another software application. However, the imagery is not linked to the maps or SIGINT. While X-IST is very powerful and innovative, it principally performs Level 1 fusion of a single source (SIGINT). It lacks an underlying database architecture for reasoning across sensors and was not designed to link different data sources within a common context.

5.1 MSIIA

An intelligence analyst who directly supports a decision maker in strategic, tactical, or mission planning, produces a report by assembling information from analysts in imagery, signals, and other areas of intelligence. For each sensor domain, there are specialized intelligence analysts who are experienced in interpreting sensor reports. It is the responsibility of the decision-oriented analyst to determine the impact of the information coming from the different intelligence sources. The MSIIA Project [12] is developing a workstation environment that enables a decision-oriented analyst to view, manage, and analyze these sources of information and, simultaneously, confer with specialized analysts. This system integrates radar sensor intelligence (RADINT), imagery (IMINT), signal intelligence (SIGINT), electronic map products, and intelligence order-of-battle databases. Because of disparate intelligence sources displayed in a common geographic context, the decision-oriented analyst can examine data collected over time and collaborate with specialists about the relationship between events detected by different sensors. Figure 11 presents a view of the MSIIA data space and associated situation assessment functions for integrating JSTARS Moving-Target-Indicator Radar, fused SIGINT, IMINT and Geographic Information Systems Data.

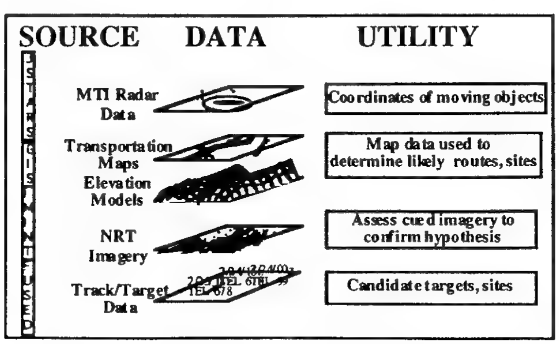


Figure 11. Multisource Information Integration

The MSIIA system combines Joint Surveillance Target Attack Radar System (Joint STARS) Moving Target Indicator Radar (MTI), SIGINT, Defense Mapping Agency (DMA) electronic map products, a variation of the Defense Intelligence Agency's Integrated Database, commercial satellite imagery, and near-real time reconnaissance imagery. A central component of this system is a commercial geographic information system, which is used to manage and display the data in a geographically registered framework.

5.2 Multi-Source Analysis and Fusion

The main goal of any information analysis and fusion system in the domain of military intelligence is to combine the available data on a specified area of interest to achieve the best possible estimate of the objects, their groupings, movements and activities. The ultimate goal of this activity is to enhanced situation understanding that will facilitate a more appropriate utilization of assets (e.g., to prioritize intelligence collection, to guide campaign planning). To exploit disparate sensor and reporting system information over a varied spatial/temporal region, several reasoning mechanisms must be employed, whether by human or machine.

The problem of intelligence analysis and fusion on the MSIIA project is exacerbated by the diversity of sensor and information types brought together in a single, integrated analysis environment. Since there are more sources of information, more information can be gleaned from it, but only if the proper reasoning mechanisms are applied. Data comes in snapshots of a continually changing world. These snapshots contain different pieces of the overall puzzle. Additionally these snapshots are generated at temporally disjoint epochs. What this means is that all the sensor information available on an object is not view able at the same spatial-temporal interval. This is caused by two phenomena. First, sensors (with the exception of Joint STARS) rarely have continuous coverage; therefore, information is gathered at discrete time intervals. The second reason is that most types of information are not being generated continually. Most objects are not going to be communicating, radar or infrared emitting, or moving continually. Thus, in most cases, the only time information is actually gathered is when the sensor is looking and the object is generating the proper signal.

Most data used by the MSIIA workstation will be received as point temporal data or spatial-temporal track data. That is, data pertaining to a specific sensor event will be tagged with a discrete time and geo-spatial location, essentially a snapshot. This information, however, limits the amount of reasoning that can be done since most objects of interest do not stand still. What this implies is that to fuse the various sensor events, reasoning about what is happening between all the snapshots is required.

5.3 Overview of the Fusion Algorithms

The MSIIA fusion mechanisms are implemented on a network of Sun UNIX workstations that house a Sybase relational database, ArcInfo Geographic Information System (GIS), and ProKappa knowledge engineering environment. All reasoning mechanisms are currently implemented in ProKappa and knowledge is represented in frames.

The knowledge based fusion process uses a constraint based reasoning model that mimics the process by which a human analyst would approach the fusion process. This approach has the advantage of allowing for explanation capabilities that can be related to the users reasoning process. There are two distinct sets of constraints that need to be satisfied in order to fuse disparate pieces of sensor information. The first constraint is that of the classification of objects. In order for two pieces of sensor/source information to be fused, they must be about the same type of object. This set of constraint satisfaction is achieved by explicit representation of the possible objects a piece of information could be on the "possible-object" slot of the individual sensor/source objects. Figure 12 presents an illustration of this representation where sensor events are shown inheriting attributes from activity type objects and sensor type objects. The bottom of the figure illustrates a particular instance of a sensor event with associated event type, date, time, location, type of object recognized and so on.

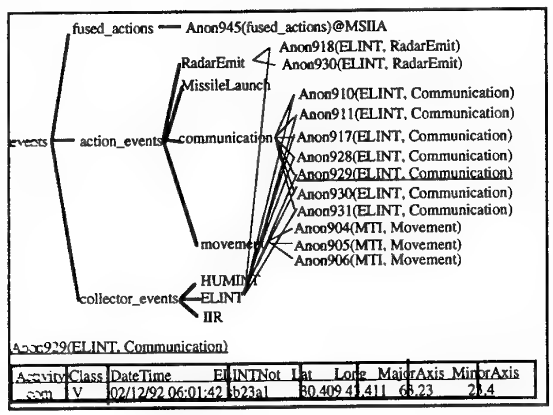


Figure 12. Sensor-/Source Object Hierarchy

Classification constraint satisfaction is accomplished by comparing the possible objects that all candidate sensor/source events can be with all other candidates. The result of this process is a set of objects that can be uniquely related to each other based on classification. With a set of objects that can be related by classification a second set of constraints that related to the relationships on the objects in space and time needs to be satisfied. Spatial-temporal reasoning is the process of analyzing objects in space and time. This is inherently difficult given the complexity of the MSIIA data space combined with the problems with representing temporal data in a two dimensional geographic space. Being able to maintain spatial-temporal relationships is a cognitively demanding task in volatile domains such as MSIIA's. Technically, implementing this type of capability is difficult since conventional relational database technology does not support complex spatial or temporal information analysis. Spatial-temporal constraint satisfaction is achieved in MSIIA by having an explicit temporal model in the fusion algorithms and spatial representation in the GIS. This

provides the functionality required to analyze the relationships between sensor/source events in space and time. The following explains how spatial-temporal constraint satisfaction is achieved.

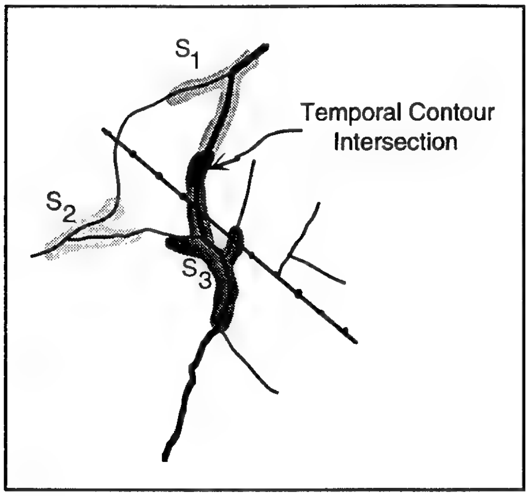


Figure 13. Mobility Contour Intersection

Given a number of sensor events that occur at different times and that can potentially be fused (i.e., they satisfy classification constraints), mobility contours for the time difference between the events are constructed. These mobility contours (generated in the GIS) represent where an object could be based on the time difference to other objects. These mobility contours are generated as a function of the "mobility characteristics" of the object in question. For example a SCUD TEL will only move on improved hard surface roads. The possible fusion occurs when there is an intersection of the mobility contours. Figure 13 presents a graphical example of a possible contour intersection. Here, three sensor events may possibly be combined, but, because of the mobility dynamics of the objects inferred from the sensor events, there is only one possible intersection, namely the intersection of S_1 and S_3. When there is an intersection of two objects a fused "binary track" is created. This becomes a new object in the fusion system and represents the relationship between two pieces of sensor/source data. Binary tracks form the foundation for assembling more complex tracks.

With a set of binary tracks which represent all possible relationships between sensor/source events, addition techniques can be employed to explore the relationship between them. The sensor/source pre-processing portion of the fusion algorithms, which are responsible for taking intelligence information (events) and populating the knowledge base, were modified to facilitate a new reasoning technique for extraction on a minimum set of unique objects and tracks from a set of data. The pre-processor constructs a set of unique one-to-one (binary) relationships between all events. These binary tracks enumerate all the possible relationships that can exist give a set of point sensor/source data under the constrains of classification and spatial-temporal mobility. Given a set of binary tracks it is then possible to construct graphs representing relations between events by using analysis techniques from graph theory to extract unique tracks of objects.

Since the fusion process is based on a model that mimics the analyst's reasoning process, this appears to yield more intuitive and credible explanations of inferences. In particular, since the fusion process is constraint based, MSIIA "justifies" its conclusions by listing the constraints that were satisfied to fuse information together. Figure 14 presents a sample of the explanation of a fused binary track. Here, two events with related classification and meeting spatial-temporal constraints are summarized for the user.

Binary track of event 187 and 462

Event 187
classification VAB
location 47.578 28.137
time 02/17/93:13.10.12

Event 462
classification VA
location 47.203 28.134
time 02/17/93:13.57.20

Time difference - 47.08
Distance Difference - 23.47
Average velocity - 29.91
Binary track confirmed by
classification and space-time

Figure 14. Sample Fusion Algorithm Explanation

Finally, MSIIA incorporates a natural language front end based on Natural Language Inc.'s COTS tool which enables the analyst to query the system for data and explanations of inferred information. Graphical displays of event sequences over time enable the user to quickly examine the inferred behavior of objects.

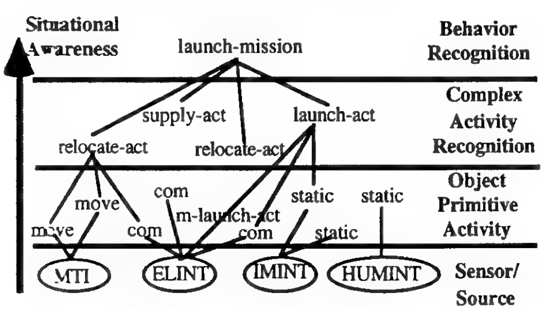


Figure 15. Levels of Recognition

Future investigations are evaluating the ability to interpret information at increasingly higher levels of abstraction. Figure 15 illustrates how sensors can recognize objects and ultimately activities, from which higher level behaviors can be inferred. As information fusion capabilities approach descriptions of enemy behavior levels,

opportunities increase for incorporation into high level campaign planning.

6. LESSONS LEARNED

A number of key lessons can be culled from the above experiences. First, Commercial off the Shelf (COTS) hardware and software has assumed strategic importance given the ability to leverage commercial investment, efficiencies, and marketplace competition [14]. This can also dramatically decrease time to field applications and maintenance tails. For example, the intelligence community's Intelink system went from concept to operational in a matter of months by replicating existing Internet functionality on classified networks. What makes COTS truly valuable are standards for information storage, processing, and exchange in order to ensure systems interoperability. This provides the added benefit of vendor independence, which enables the government to take advantage of marketplace competition assuming there is no vendor monopoly. Distributed object management will help further this trend as third party vendors become able to add value to products without having to first develop full-featured offerings.

Finally, unlike traditional multi-year or multi-decade acquisition cycles where a formal process of requirements analysis through acquisition and finally logistical support is rigidly followed, the pace of political, military doctrine, and technology change underscore the importance of a collaborative, evolutionary approach to acquisition. By this we mean that multifunctional teams, from technology providers to end users, are assembled to rapidly deploy, in a phased approach, fielded capabilities which are refined to meet operational requirements through direct interaction with and involvement of end users. In cases where legacy systems are too complex or expensive to re-engineer, object request brokers can serve as an important element in supporting interoperability. Lastly, in any system that will be developed for tasks as complex and involving as many uncertainties as crisis action planning, truly powerful systems will only be possible when we find effective mechanisms that utilize both machine and human knowledge synergistically to enhance overall system performance.

7. A VISION FOR THE FUTURE

Despite formidable mission planning systems such as CTAPS, there exists no current set of collaborative, campaign and theater-level mission planning and battle coordination tools to support joint or international operations. This is exacerbated by the lack of a common information infrastructure at multiple levels (including data element standards, network protocols, security services, and user applications). This has resulted in limited system interoperability which minimizes possible information sharing and real-time coordination of joint and multinational teams. This limits joint coalition forces from effective, real-time resource reallocation and rescheduling and results in decreased resource utilization (increased cost) and increased force risk (e.g., unthwarted enemy threats, fratricide).

Figure 16 shows a vision for knowledge-based, distributed, collaborative planning to support the Joint Task Force Commander, a notional integrated view of capabilities described in this article. These include:

- Knowledge-based planning and scheduling aids, at the campaign and theater level, which exploit techniques such as hierarchical planning, case-based reasoning (e.g., about historical/enemy battles) and knowledge-based simulation of friendly and enemy forces.

- Multisource correlation/fusion and enemy behavior learning, recognition and prediction using statistical, pattern-recognition, and knowledge-based techniques.

- Highly interactive, intuitive, and intelligent human-computer interfaces that support multidimensional situation analysis and course of action visualization

- Collaborative tools that enable not only information sharing but virtual collaboration among users.

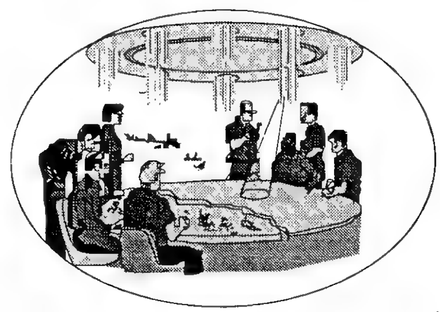


Figure 16. Vision for Intelligent, Distributed, Collaborative Planning

7.1 Capability/Functionality

This vision comprises three key operational facilities: an intelligent and intuitive mission planning interface for joint and multinational use, a set of collaborative, knowledge based mission planning tools, and an information infrastructure that will enable the above.

First, automated multisensor selection and/or multisensor fusion in the context of an intelligent and intuitive display will enable a senior intelligence officer and perhaps even the commander to interactively perform situation assessment. Importantly, the human-machine interface will provide multimedia, multilingual and multiparty interaction to support collaboration with coalition forces. A user-adaptive interface will provide rapidly customizable views for specific task functions, including browsing, search, and visualization of real-time as well as historical information. This will include summary views (e.g., of fused tracks, of overall characteristics of a class of objects) which will allow rapid access to supporting details for further analysis or verification.

Second, the underlying systems will provide a shared set of intelligent mission planning and scheduling tools. These will facilitate decision making through use of multiple capabilities including:

- Access to historical missions/battles to analyze enemy propensities for response/attack
- Intelligent agents to discover and filter critical information and patterns
- Real-time simulation of friendly and enemy forces (using real data mixed with simulated agents) to support rapid what-if analysis
- Knowledge based planning and scheduling tools that take into account factors such as current intelligence, weapons characteristics, logistics constraints, weather conditions, and objectives and strategies, to facilitate decision making through recommendations of alternative courses of action
- Embedded, on-line intelligent trainers for learning advanced system features in non-crisis periods as well as providing on-line task assistance during operations

The aim of all of these facilities will be to increase the cognitive power and efficiency of the decision maker.

To support the above requires a critical third element: mechanisms for bridging the gaps between existing stovepipe systems to support rapid integration of joint and coalition systems in crisis situations. This includes not only access to heterogeneous databases (e.g., intelligence, operations, and logistics data) but also interoperability of higher level application tools (e.g., requirements management, target nominations, force allocation). Modules would be integrated using open systems approaches (e.g., object-request broker standards, messaging and directory services standards) which support evolutionary development of new facilities to facilitate technology transfer.

7.2 Technology Needed

Several technologies are key to enabling the above facilities. First, we require technologies to support adaptive, intuitive user interaction. These include self-adaptive input/output devices, multilingual speech recognition and generation, multimedia presentation planning, natural language dialogue management, information summarization, and virtual displays [15].

To support intelligent decision support, we require a host of technologies such as real-time knowledge based simulation and planning tools. This will result in severe computational challenges, for example, to support large object-oriented and knowledge-intensive simulations (e.g., simulating hundreds of thousands of battlefield objects). The collaborative nature of the tools will require advances in workflow management and intelligent routing (e.g., to support joint and multinational tasking, dissemination of indications and warnings).

Finally, several infrastructure advances are required to facilitate information sharing and collaborative planning. These include scaling up approaches such as the Object Request Broker (ORB) to integrate legacy systems and support rapid integration of and evolution toward new capabilities. Multilevel security will clearly be an issue given the number and types of partners likely to be interacting using such a system and their differing information needs. Communications requirements and complexities will require more sophisticated approaches to network and systems management (e.g., active performance management, knowledge based fault detection, diagnosis, repair). Communicators will demand real-time video teleconferencing as well as application and multimedia information sharing (e.g., maps, imagery) which will likely require gigabit and terabit networking but also advances in compression techniques and wireless technologies to support the soldier, airman, and seaman in the field.

To make progress toward the outlined vision requires not only coordinated technological investment by NATO member nations, but also commitment to more toward common architectures. Action should include:

1. Sharing lessons learned with NATO member nations.
2. Building a common NATO infrastructure by exploiting advances in distributed object technology to set the stage for future systems integration.
3. Establish a working group to forge a common vision for distributed, collaborative planning systems that can foster user pull and international partnering to move in this direction.

8. CONCLUSION

Global geo-political, economic, and military acquisition changes are driving a fundamental questioning of both what is needed to support national and global security and how best to provide that. An increasing trend toward interdependent political, economic, and military systems has focused attention on the need for improved joint service and international systems. We currently lack of a set of collaborative, integrated, interservice and international campaign and theater-level mission planning and battle coordination tools. This is exacerbated by the lack of a common information infrastructure at multiple levels (including data element standards, network protocols, security services, and user applications).

This has resulted in limited system interoperability which minimizes possible information sharing and real-time coordination of joint and multinational efforts. This limits joint coalition forces from effective, real-time resource reallocation and rescheduling and results in decreased resource utilization (increased cost) and increased force risk (e.g., unthwarted enemy threats, fratricide). This article outlines the emerging role of distributed object management, and forthcoming distributed, collaborative force deployment and employment tools that, together with a new approach to procurement and a vision for the future, can help address the serious existing shortfalls in interoperability, functionality, and systems acquisition.

9. REFERENCES

1. A National Security Strategy of Engagement and Enlargement, July 1994. The White House (ISBN 0-16-045153-1)

2. Widnall, S. E.1995. State of the Air Force. *The Hansconian*, January 6, 1995.

3. Knowledge Based Guidance and Control Functions. 1994. Draft AGARD Advisory Report from Working Group 11 of the Guidance and Control Panel of NATO/AGARD.

4. Brando, T. J. December, 1994. DOMIS Implementation of CTAPS Functionality Using Orbix. MITRE Paper 94-B0000287.

5. Thomas J. Brando and Myra Jean Prelle, October, 1994. "DOMIS Project Experience Migrating Legacy Systems to CORBA Environments", OOPSLA-94 Workshop Is CORBA Ready for Duty?.

6. Walker, E. and Cross, S. (eds) to appear. Proceedings of the ARPA-Rome Laboratory Planning Initiative Annual Workshop, Morgan-Kaufmann Publishers, June 1994.

7. Mulvehill, A. 1994, September 17. Distributed Collaborative Planning Tools: JWID-94 Evaluation Results. Unpublished Report. MITRE Corporation, Bedford, MA 01730.

8. Cross, S., Roberts, D., Mulvehill, A. and Sears, J. A. 1994. Case-based Reasoning Applied to a Force Generation Decision Aid. Eighth International Symposium on Methodologies for Intelligent Systems, Charlotte, N.C., October 16-19, 1994.

9. Coley, S. M., Connolly, D., Koton, P. K., McAlpin, S. and Mulvehill, A. M. SMARTPLAN: A Case Based Resource Allocation and Scheduling System: Final Report, MITRE MTR-11270, The MITRE Corporation, Bedford, MA 01730.

10. The Air Campaign Planning Tool: An Introduction. Draft. 1 July 1993. HQ USAF/XOCC -- Checkmate.

11. ACPT - The Air Campaign Planning Tool. ISX Corporation, Westlake Village, CA. White paper.

12. Hansen, S and Nakamoto, G. June, 1993. "Multi-Source Integration and Intelligence Analysis." Joint Directors of Laboratories Data Fusion Symposium. Vol I , 171-182.

13. Joint Directors of Laboratories Data Fusion Subpanel [1992]

14. Horowitz, B. M. 1993. *Strategic Buying for the Future: Opportunities for Innovation in Government Electronics System Acquisition.* Libey Publishing Inc.: Washington, D.C.

15. Maybury, M. T. editor. 1993. *Intelligent Multimedia Interfaces*. AAAI/MIT Press.

10. ACKNOWLEDGMENTS

This paper presents an integrated view of multiple on-going activities involving many individuals and multiple organizations (e.g., ESC, ARPA, RL, MITRE). Particular thanks goes to Steve Hansen who contributed much of the information fusion section. I also thank Dr. Myra Prelle, Thomas Brando, Alice Mulvehill, Alice Schafer, Penny Chase whose CTAPS and ForMAT work I summarize. Thanks also to Cliff Miller for sharing his domain expertise and guidance. Further appreciation goes to Glen Nakamoto, Steve Cross, Nort Fowler, Don Roberts, Lou Hoebel, Steve Huffman, Ed Green, Bob Nesbit, Bob Frost, and Ed Thompson, who are responsible for several programs described in this article. I thank Lt Col Douglas Ford for feedback on the final draft. Finally, I thank Don Neuman for encouraging me to write this article. Any omissions or errors are my own.

11. GLOSSARY

ACPT	Air Campaign Planning Tool
AFMSS	Air Force Mission Support System
AOC	Air Operations Center
APS	Advanced Planning System
ARPA	Advanced Research Project Agency
ATO	Air Tasking Order
ATD	Advanced Technology Demonstration
COTS	Commercial Off the Shelf
COA	Courses of Action
CTAPS	Contingency Theater Automated Planning System
CINC	Commander-in-Chief
CAFMS	Computer Assisted Force Management System
C^4I	Command, Control, Communications, Computers, and Intelligence
DART	Dynamic Analysis and Replanning Tool
DMA	Defense Mapping Agency
DODIIS	Department of Defense Intelligence Information System
ESC	US Air Force Electronic Systems Center
ForMAT	Force Module Analysis and Management Tool
GCCS	Global Command and Control System
GIS	Geographic Information System
GLAD	Geographical Logistics Awareness Display
HUMINT	Human Intelligence
IMINT	Imagery Intelligence
IFD	Integrated Feasibility Demonstration
JCS	Joint Chiefs of Staff
JDISS	Joint Deployable Intelligence Support System
JDL	Joint Directors of Laboratories
JFACC	Joint Force Air Component Commander
JFLCC	Joint Force Land Component Commander
JFMCC	Joint Force Maritime Component Commander
JFSOCC	Joint Force Special Operations Component Commander

Joint STARS	Joint Surveillance Target Attack Radar System
JTF	Joint Task Force
JWICS	Joint Worldwide Intelligence Communications System
JWID	Joint Warrior Interoperability Demonstration
JMAPS	Joint Message Analysis and Preparation System
MOIE	Mission Oriented Investigation and Experimentation
MSIIA	Multisource Intelligence Integration and Analysis
MTI	Moving Target Indicator
RL	US Air Force Rome Laboratory
RAAP	Rapid Application of Air Power
SIGINT	Signals Intelligence
SOCAP	SIPE-II Operational Crisis Action Planner
SIGINT	Signals Intelligence
TARGET	Theater Analysis, Replanning and Graphical Execution Toolbox (IFD-3)
TPFDD	Time Phased Force Deployment Data
USTRANSCOM	United States Transportation Command
X-IST	Extended Intelligence Support Terminal

Functional Development and field test of CASSY - a knowledge-based cockpit assistant system

R. Onken
Universität der Bundeswehr München
Werner-Heisenberg-Weg 39
D-85577 Neubiberg, Germany

SUMMARY

This paper presents the functional concept and development of the cockpit assistant system CASSY, based on basic requirements for effective man/machine interaction. This system was developed in order to enhance flight safety and mission effectiveness. The time has come that cockpit systems will no longer be designed on a vague basis of specifications. The advances in technology provide the necessary basis to systematically reflect requirements for human-centered automation into clear-cut specifications and system design. CASSY is developed as a knowledge-based system. It has been extensively tested in flight simulators as well as field tested in the ATTAS (Advanced Technologies Test Aircraft System) of the DLR. Some of the results of these flight trials will be presented in this paper. The development was conducted by the University of the German Armed Forces, Munich with some support by DASA.

1. INTRODUCTION

Advances in electronics and computer technology have a profound effect on modern aircraft. On the flight deck electronic displays and computer controlled avionic devices are common features. Powerful computer technology is involved in the signal processing to generate display formats and to transmit signals to and from control devices and the remaining avionic components.

However, the crew station as we know it today is undergoing an even more profound change, which will allow to make use of abstract human-like knowledge and reasoning:

- to understand the abstract goals and subgoals of a flight mission,
- to assess needed information about mission, aircraft environment and aircraft system
- to interprete the flight situation in the light of the goals of the flight mission
- to detect pilots' intent and possible errors.
- to support necessary replanning and decision making,
- to know which information the crew needs and how to present it to the crew in an effective way

Thriving for this change in the crew station means thriving for increase of mission effectiveness. It also means increase of safety. It is known since long that erratic human behaviour is the main contributing factor in about 75% and more of all accidents in civil aviation. It can be claimed that these human failures are caused by some kind of pilot overtaxing, either clearly realized by the pilot as an overdemand or not even noticed as such until it is too late. In this context, overtaxing is considered as describing the situation when potential resulting human failure is imminent because of inherent human deficiencies in sensory, cognitive and effectory capabilities and performance. New types of latent overtaxing-prone situations appeared with the advancement of automation in the aircraft cockpit, in particular with respect to failures in situation awareness. Recent accidents of civil aircraft with state-of-the-art cockpit automation provided some evidence for this particular consequence. This does not mean that automation as such is causing this mismatch. It is the way as automation is used for flight safety and mission effectiveness and to what extent and by which interaction guidelines autonomous machine functionality is exploited for effective function sharing between man and machine, not to replace the human pilot rather than to support the pilot along the ideas of human-centered automation.

To think about the next generation of cockpit-avionics means to realize which role comes up to the crew in the light of the aforementioned machine capabilities and which basic requirements for automation have to be met to comply with increased flight safety and mission effectiveness demands. Function allocation to aircrew and machine has to be reconsidered.

Function allocation is not such an easy task as it might appear at the first glance. The assignment of functions or part functions to be allocated to the machine in one or the other way is driven by the design objective to reduce crew workload, by letting the machine do what it can do or what it can do better. Therefore, technical feasibility often times seemed to be sufficient reason to automate certain functions in whatever type of allocation, hoping for some kind of overall system improvement and crew workload reduction. To let the machine do what it can do better, however, might lead accidentally to allocations of certain functions to the machine, of which part functions could be carried out much better by the crew than by the machine or of which part functions should be carried out by the crew to keep the pilot in the loop.

Figure 1 illustrates, which are the functions the crew is trained to perfom, compared to those functions allocated to the aircraft systems. There are those functions, which have to be activated by the crew and thereby to be allocated in

Paper presented at the Mission Systems Panel of AGARD and the Consultant and Exchange Program of AGARD held in Madrid, Spain, on 6-7 November 1995; Châtillon, France, 9-10 November 1995; and NASA AMES Research Centre, California, USA, 16-17 November 1995 and published in LS-200.

order to carry out certain tasks on request and in place of the crew and there are also those machine functions (usually not considered under the aspect of function allocation), which are permanently turned on like the basic cockpit instrumentation and actuator machinery for power amplification.

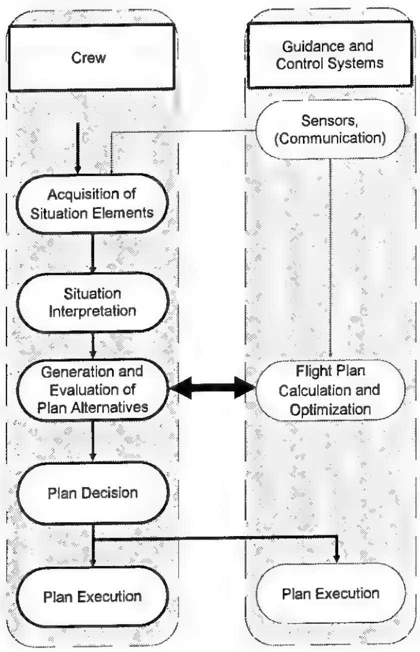


Figure 1. Flight Guidance and Control Today

Not but recently it became evident by a number of accidents that the principle of function allocation as it is deployed so far might lead to problems. Automated functions were not sufficiently scrutinized as to their impact on the overall mission effectiveness. There is a steadily increasing number of permanently activated, more complex but hardly intelligent (knowledgeable) machine background functions the crew has to keep track of and at the same time there are increasing numbers of options of machine functions at the disposal of the crew and to be controlled by the crew. This might become very complex to control in view of the fact that at the same time the crew should be ready to take over all part functions at any time which were discontinued to be covered by the machine for any reason.

This results, for instance, in two major concerns with regard to function allocations to the machine:

1. Functions permanently turned on for system configuration, for instance, are often event-driven and working in the background. Their operation and the resulting changes in the aircraft range of maneuverability and pertinent consequences might not be aware to the crew and might lead to overtaxing in crew situation assessment.
2. Functions intentionally turned on by the crew might unexpectedly demand for too much attention by the crew because of complex handling.

Consequently, it is not surprising that increased automation without thinking about new ways of function allocation implies increased potential of new types of crew overtaxing and resulting human failures, i.e. mission hazards.

Therefore, new ways of automation and functional share between aircrew and machine have to be established. An extremely important first step is a top down structuring of design requirements, taking into account the aspects of human-centered automation in a systematic way. These basic requirements will be formulated in the first part of the paper. To describe this in some more detail is one of the main purposes of this paper, taking the most advanced cockpit system on state-of-the-art flight decks, the flight management system (FMS), as an example of current automation dilemma to be overcome.

On the basis of the top down structure of requirements as mentioned before, machine functions can be specified, which will really serve the mission goals. This leads to the second part of the paper where the development of the Cockpit ASsistant SYstem CASSY and the system field tests will be described.

2. THE FLIGHT MANAGEMENT SYSTEM

In an US investigation on aspects of the interaction between man and machine in the cockpit [Wise et al., 93] some of the problems with the FMS were highlighted. Other investigations came to similar results.The FMS receives information about the actual flight, including data about the destination, the flight plan to the destination with way points and altitudes, weather information and weight of load. When these informations are keyed into the system by the crew, which can become a significant interactive effort, the FMS function can be initialised. From then on the aircraft can fly autonomously unless no changes of inputs have to be keyed in because of unexpected encounters in the overall flight situation. The conclusion of the investigation was that the pilots like to make use of the automatic functionality of the FMS, however they run into difficulties in time-critical situations with unforeseen constraint impacts like new ATC instructions. For these situations there is not sufficient spare time for the necessary inputs and the interpretation of computational results as delivered by the FMS. These are the situations when the pilots might be left on their own [Wiener, 89; Amalberti, 92] with questions like

what is it doing?
why did it do that?
what will it do next? or
how did it ever get in that mode?

Thus, the FMS is usually turned off just at situations when the pilots starvingly look for assistance [Heldt, 93].These obvious deficiencies clearly indicate that the FMS at this stage of automation is not advanced enough to principally avoid compromising flight safety. Certain principles of securing aircrew situation awareness have come somewhat out of sight. Therefore, it is time, now, to reconsider the basic requirements for machine support in the cockpit, in particular regarding situation assesssment tasks of the crew including sensory and information processing functions.

3. BASIC REQUIREMENTS FOR COCKPIT AUTOMATION

There are a great number of well-formulated requirements

at hand for man-machine interaction in the cockpit, including those for "human-centered automation" [Billings, 91]. However, in order to systematically merge future automation into what is really wanted with regard to flight safety and mission effectiveness, it should be possible to assess how much certain individual requirements from the long list of existing ones contribute to the design goals, and what are the interdependencies. This is extremely important, in particular, when trade-offs are necessary for any reason and priorities are to be defined.

Therefore, a top down structure of as few as possible basic requirements is needed which will be described in the following, easing the engineering task of converting the requirements into a technical product. In order to resolve this problem, the objective of automation has to be redefined in general terms: Simply, the objective is to avoid overtaxing of the cockpit crew. That means that the demands on the cockpit crew have to be kept on a normal level for all situations and situation-dependent tasks. The aircrew task domains to be considered are flight control, navigation, communication and system handling and the task categories under these domains are

- situation assessment,
- planning and decision making and
- plan execution

For these task categories the following priority list in terms of a hierarchy of two levels of basic requirements can be established [Onken, 93]. These requirements are essentially equivalent to the requirements for human-centered automation as stated in [Billings, 91], however, they are structured differently in favor of the engineering point of view with respect of mechanisation. They can be formulated as stated in the following:

1. To avoid overcharge of the crew in situation assessment, the top requirement
   BASIC REQUIREMENT (1) should be met, i.e.:
   Within the presentation of the full picture of the flight situation it must be ensured that the attention of the cockpit crew is guided towards the objectively most urgent task or subtask of that situation.

2. In order to avoid or decrease overcharge of the crew in planning/decision making and plan execution, as a subordinate requirement
   BASIC REQUIREMENT (2) can be formulated:
   If basic requirement (1) is met, and if there still comes up a situation with overcharge of the cockpit crew (in planning or plan execution), then this situation has to be transferred - by use of technical means - into a situation which can be handled by the crew in a normal manner.

This particular top down formulation of requirements for human-centered automation distinctly makes clear that whatever technical specifications are made for systems in support for the cockpit crew, they are questionable if the specification for the situation assessment capability of the support system (Basic requirement (1)), including the assessment of the crew's situation, is too neglectful and sloppy. How can the support system work on directing the crew's attention, if it cannot assess the global situation on its own? If the system is not able to understand the underlying situation, it might work on the basis of wrong assumptions! Thus, if the specification fails with regard to basic requirement (1), this cannot be compensated by whatever automated support designed to comply with requirement (2) only.Unfortunately, this inadequacy by disregarding basic requirement (1) usually was the case in the past, essentially because the technical means were not available for comprehensive situation assessment by the machine. Prevention of overcharges concerning situation assessment was not worked into the specification in the systematic manner as it is suggested by basic requirement (1).

Basic requirement (1), in fact, compulsorily leads to the full set of specifications which in turn can be used to verify human-centered automation design.

4. HOW TO APPLY THE BASIC REQUIREMENTS FOR SYSTEM DEVELOPMENT

Obviously, according to basic requirement (1), there is the main issue to carefully specify the situation assessment part of the machine functions. The picture of the flight situation as generated by the machine should cover all aspects which are also to be considered as situational aspects by the cockpit crew. Moreover, it would be most desirable, if the machine picture would be even more comprehensive and more accurate. This is already feasible today in certain aspects. In principle, thereby compliance with basic requirement (1) can be accomplished with the technology at hand today. In essence, the capability of situation assessment is to be incorporated in terms of corresponding functions in the machine part of the man/machine system in parallel to those of the cockpit crew (figure 2). As part of the situation assessment, the machine is attentively watching the cockpit crew's state and activity, thereby having the full picture including the crew situation.

This is the basis for cooperative automation in order that the cockpit crew's attention can be guided towards the objectively most urgent task or subtask of the actual situation. It becomes evident at this point that instead of allocating functions either on the machine side or the crew side once and for all times, all functions necessary to fly the aircraft are not only inherent crew functions but also functions which the machine should be capable to perform. All of them are operative in parallel unless effector actions are to be executed. Thus, there is no conflict with the principle that it is generally up to the crew to make the final decision about whether to accept action recommendations of the machine or to follow their own ideas. We call this the situation-dependent functional share of man and machine as partners. Partnership means that the capabilities of the partners are similar, but not necessarily identical. Partnership demands for effective dialogue. According to basic requirement (1), the presentation of the full picture of the situation has to be shaped in a way that the crew's attention is guided by the presentation only if necessary. In addition, the crew should also be able to talk to the machine partner like the crew communicates among each other. Therefore, the key specifications for the development of new generations of cockpit automation, in summary, concern both

- comprehensive machine knowledge of the actual flight situation and
- efficient communication between crew and machine, based on situation

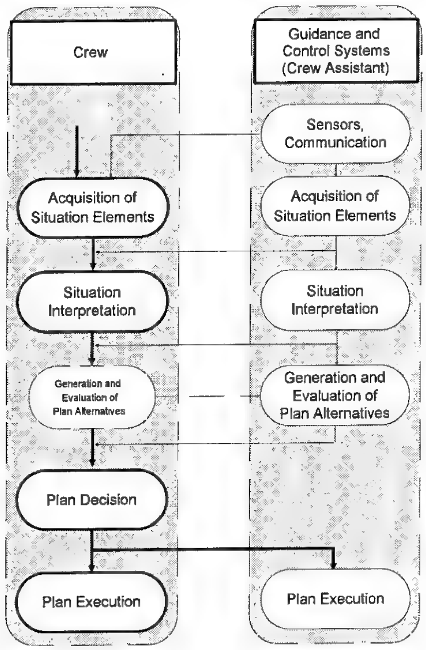


Figure 2. Flight Guidance and Control Upcoming

knowledge and new dialogue technology. How can the machine knowledge about the actual flight situation be established in order to meet these specifications? Both advanced techniques for structured knowledge representation and information processing based on advanced sensor technology (e.g. voice recognition and computer vision) allow for generating the knowledge base which includes about all static and dynamic situation elements the cockpit crew may be aware of and possibly even more than that. The task-related situation elements are concerned as well as the elements pertinent to the main players like the world surrounding the aircraft, the aircraft itself and, probably most important, the cockpit crew (figure 3).

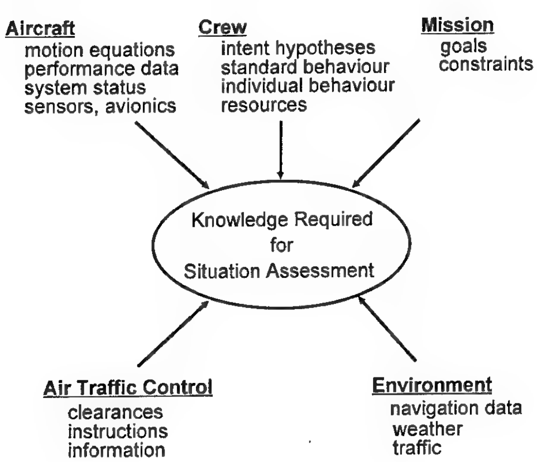


Figure 3

The knowledge about the cockpit crew is crucial. Objective knowledge about the crew can be of paramount value. On the one hand, the machine might have a better picture of the pilot's status than the pilot himself, in particular in situations of imminent overcharge. On the other hand, machine knowledge about the crew is the basis for crew-adapted assistance. The machine cannot assist in an efficient way, if it does not sufficiently understand the cockpit crew's activities and corresponding needs. In its most advanced elaboration the knowledge about the cockpit crew comprises models of the physical and mental resources as well as behavioural models (see figure 4).

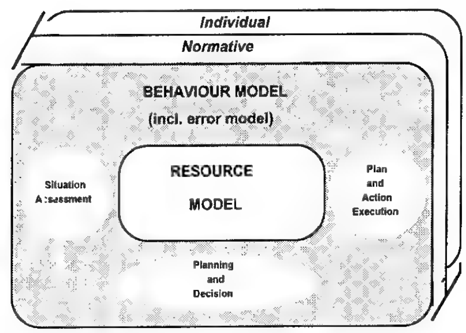


Figure 4. Model for Cockpit Crew

Thereby, the crew behaviour for situation assessment, planning and plan execution is to be modelled for normative behaviour as well as individual behaviour. The knowledge about the crew member's individual behaviour has to be learned on-line by the machine. Modelling of the error behaviour is another important behavioural aspect to be covered. Crew action modelling should not be confined to activities with hands and feet, also eye and head motion as well as voice activity contain important information, also with regard to efficient communication management between machine and crew.

In summary, chapter 3 and 4 have outlined the main guidelines, in little depth though, which are to be followed as closely as possible in order to warrant human-centered automation. These guidelines can easily be formulated as system design specifications.

5. THE DEVELOPMENT OF THE COCKPIT ASSISTANT SYSTEM *CASSY*

5.1 Main structure of CASSY

The following description of the cockpit assistant system-CASSY for civil transport aircraft will present a possible solution to comply with the dicussed ideas. CASSY was developed at the Universität der Bundeswehr München (UniBwM), including a certain amount of support by DASA.

The main structure of CASSY is shown in figure 5. All situational elements of the entire flight situation, including mission, aircraft, systems, environment, and crew aspects, are stored in a central object-oriented representation, not explicitly depicted in figure 5. A specific communication module, the **Dialogue Manager** [Gerlach and Onken, 93] is responsible for picking the relevant advisory patterns from the central situation representation and for coordinat-

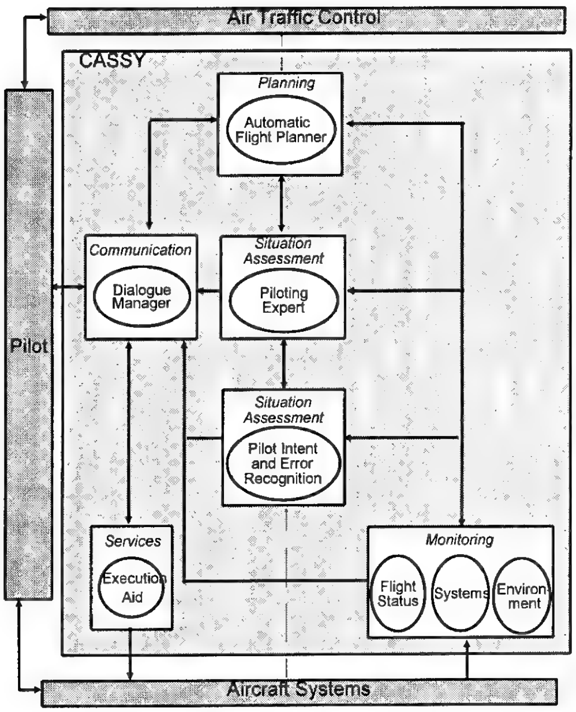


Figure 5. CASSY structure

ing their output to the crew via speech and/or graphic display. Vice versa the Dialogue Manager picks up the inputs of the crew, if there are any, and directs them to the respective module of the assistant. This is done via speech recognition. Since there should not be a permanent need for the pilot to tell his electronic partner about his state or his intentions, CASSY has to gain most of this information by observing and knowledgeably interpreting the pilot's actions. The **Automatic Flight Planner** [Prévôt and Onken, 93] generates a complete 3-D/4-D flight plan. This is done autonomously or interactively with the crew, which depends on the situation. According to the generated flight plan, the **Piloting Expert** [Ruckdeschel and Onken, 94] elaborates the expected pilot action patterns on the basis of pilot modeling such that the **Pilot Intent and Error Recognition** [Wittig, 94] can compare this expected behaviour to the actual behaviour of the crew. Thereby, it identfies discrepancies in behaviour and their reasons. There are three possible reasons for this:

a) Pilot error:
   The pilot deviates from the objectively correct expectation of behaviour, derived by CASSY.
b) Temporary discrepancy of pilot intent:
   Events, which CASSY does not know yet, cause the pilot to deviate from the behaviour expected by CASSY.
c) Machine error:
   An inappropriate or erroneous modeling or information processing within the machine (including CASSY) leads to an objectively wrong expectation of pilot behaviour by CASSY.

In case of a pilot error, a warning or hint is given to the pilot to correct the error. In order to cope with the temporary discrepancy of pilot intent, CASSY tries to figure out the intention, modify the flight plan, accordingly, and elaborate the consistent expected behaviour. Machine error should not appear, but realistically it sometimes does and must be considered. The errors are less serious, when they can be detected easily, be recovered with very few commands and have no safety critical consequences. Therefore, capabilities for in-flight restart of the system or recovery from erroneous states have to be provided as an important functionality.

In addition to the situation assessment concerning pilot behaviour, conflicts with the flight plan are detected autonomously and conflict hints are given or replanning is initiated to solve the problem. In the conflict case CASSY decides whether to initiate an interactive replanning or a completely autonomous replanning, depending on the available time and human resources. If the pilot decides to initiate planning, the amount of inputs he gives is up to him. The assessed situation is permanently shown with respect to the current flight plan on the display in heading-up or plan-mode. When no problems occur and everything is working properly, the crew does not become aware of CASSY activities other than the appropriate presentation of the flight situation.The planning and decision making assistance includes:

- autonomous or interactive generation and evaluation of routings or routing alternatives and trajectory profiles for the complete flight or local portions of the flight
- evaluation and selection of alternate airports and emergency fields
- prediction of the remaining flight portions, when ATC redirects the aircraft or the pilot intentionally deviates from the plan.

The **monitoring capabilities** include

- monitoring of the pilot actions with regard to nominal flight plan values, i.e. altitude, heading/track, vertical velocity, speeds and configuration management, e.g. flaps, gear, spoiler and radio navigation settings
- monitoring of violations of specific danger boundaries, including minimum safe altitudes, stall and maximum operating speeds and thrust limits.

Basic services are provided for

- configuration management by speech input,
- approach briefings, departure, approach and profile charts generated from the actual flight plan on request,
- performance and navigation calculations on request.

In the following, some of the CASSY modules like the Automatic Flight Planner, the Piloting Expert and the Dialogue Manager are described in some more detail.

5.2 Automatic Flight Planner (AFP)

Planning is a process of problem solving with the task to find a sequence of state transition operators which transform an initial condition into a desired goal state [Wilensky, 83]. The ordinary tasks of an on-board flight planning module include updating the flight plan, when a checkpoint is passed and/or in time-discret intervals to re-calculate headings, speeds and times of overflight. Another point is coping with all kinds of ATC instructions, i.e. adjusting the flight plan, accordingly. Most of the time altitude or speed instructions require only more or less small modifications of the trajectory-profile. A new lateral instruction, espe-

cially a radar vector, however, demands more intelligence to insert it into the flight plan, because the termination of the vectoring is unknown. Planning tasks which arise of more unusual events include those like local diversions for avoidance of bad weather areas, selection of less frequented or more economical standard routes or flight levels and the appropriate reaction to a closed destination airport. In addition, the aircraft system status and the available aircraft performance might change during the flight, in the wake of which local or global replanning could be required. Especially in major problems, such as engine fire or loss of cabine pressure, when crew workload is highest, a planning system must be able to issue a new flight plan to the next alternate airport or emergency field.

The active flight plan is the basis for providing the assisting functions of almost all modules of CASSY. The first flight plan before getting airborne can be determined by the AFP by entering the precomputed routing of the company dispatcher and making the updates of data inputs if necessary. A new flight plan is only developed by the AFP, when the flight situation demands for a modification of the current flight plan. To determine such a necessity is part of the situation assessment component of the AFP. Based on this its replanning processes generate the new flight plan.

Therefore, the AFP has to perform two different tasks in principle:

- situation assessment for flight plan conflict detection and interpretation, including the preparation of messages to inform the crew as well as autonomous activation of necessary replanning processes.
- a variety of replanning processes to adjust the flight plan to the new situation.

The conflict module reports its result, e.g. replanning necessities to the planning supervisor. This supervisor is realized as coordination automata, controlling the replanning processes by activating the needed submodules step by step. An overview of the central information flow between conflict detection and flight planning of the AFP is given in figure 6.

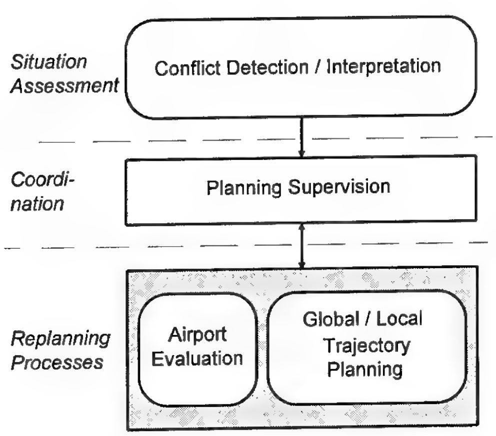


Figure 6. Central information flow

The submodules and their specific tasks as well as knowledge representation are explained in the following in more detail.

5.3 Situation Assessment: Conflict Detection and Interpretation

The situation assessment element focusses on improving the pilot's situation awareness as to flight plan executability and is responsible for autonomous activation of replanning processes. Therefore, the submodule examines the flight plan for possible conflicts. If in at least one of the flight portions ahead a potential problem is detected, the pilot crew will be informed and the relevant replanning processes are activated. More serious conflicts, which probably become safety critical, have to be solved immediately. Independently, replanning processes can also be activated on request of the pilot crew. The process includes three phases:

- Conflict detection
- Classification and assignment of conflict zones
- Interpretation

The conflict knowledge base consists of conflict hypotheses in combination with an abstract description of risk categories. During the detection phase the flight plan is examined for any occurence of a predefined conflict hypothesis. When a potential conflict is detected, instantiations are made in a conflict and danger model consisting of four conflict zones along the flight plan portions ahead:

- information zone:
  information of the pilot crew about the detected conflict is required; replanning processes for conflict resolution are in stand by for requests
- resolution zone:
  conflict resolution is initiated autonomusly by activating the selected replanning processes
- approach zone:
  the conflict zone will soon be reached; replanning only on request
- conflict zone:
  the conflict is active

These zones establish fundamental information for other CASSY core modules to cope with the detected conflict.In the interpretation process, the situation is associated to the conflict zones and the required actions are initiated. The situation dependent conflict hypotheses as part of the conflict knowledge base include all information which is needed by the AFP to activate the relevant replanning processes.

5.4 Flight Planning Capabilities

The following functions are available for replanning and decision making

- planning supervision
- information acquisition
- airport selection
  - alternate airports
  - emergency fields
- trajectory planning
  - trajectory specification
  - route planning for standard / non-standard rerouting and local diversions
  - radar vectoring estimation

- profile planning / optimization
- flight plan updating

Figure 7 zooms out the planning functionalties of figure 6, which will be looked at more closely in the following paragraphs.

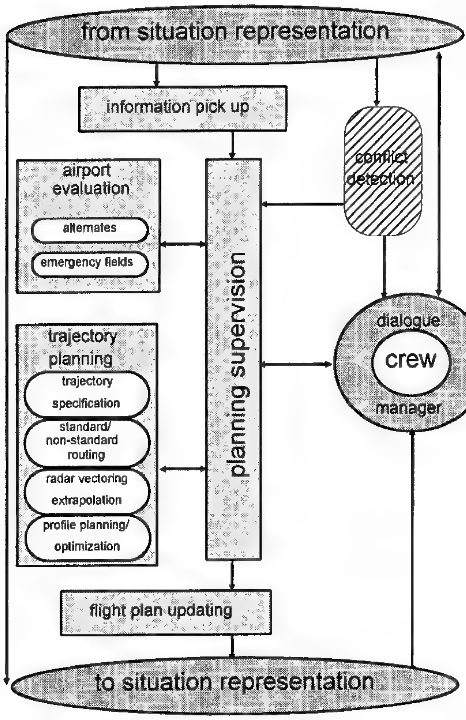


Figure 7. Structure of the Automatic Flight Planner

To ease understanding of the implemented replanning functions, it is necessary to explain the fundamental principle of flight planning and decision making followed in the AFP.

5.4.1 Principle of Flight Planning and Decision Making

Planning and decision making are strongly linked to each other. The basis for determining a sequence of actions to reach the goal state, i.e. planning, is the decision for an alternative at each of the branching nodes in the problem space. So, the principle is to generate alternatives, evaluate them and make a selection decision for one of them. The pilot crew is involved in the decision making process. The evaluation of each alternative is based on knowledge about the desired goal state. In the problem space of civil IFR (Instrument Flight Rules) flights there are generally accepted criteria and measures for evaluation in a given situation. These criteria include boundaries of flight safety and economics. Since the situation in the real world can hardly be determined exactly, uncertainty has to be taken into account [Prevot, 91]. Individual preferences have to be considered, too, to come as close as possible to the ranking scale the pilot crew would use. Obviously a great amount of partly fuzzy or uncertain criteria has to be evaluated to draw a conclusion about recommendation or rejection of an alternative. For the tasks of flight planning and decision making with CASSY a special hierarchical evaluation principle has been developed and applied. It enables structuring the criteria in clearly arranged levels and groups. It is also possible to switch the active evaluation diagrams, i.e. criteria, measures and weights, to adapt the evaluation process to specific tasks and/or individual pilot preferences. For illustration purposes an excerpt of a fundamental evaluation diagram for a decision in the scope of an IFR flight is depicted in figure 8:

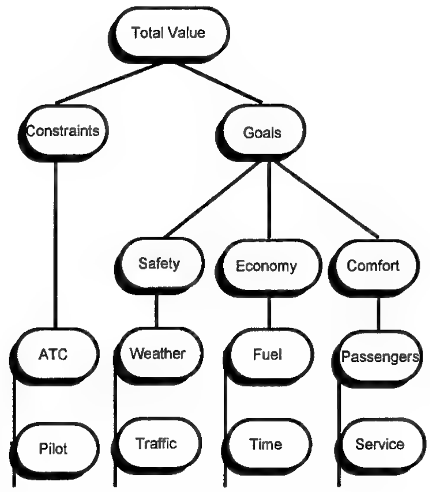


Figure 8. Evaluation diagram (excerpt)

The value of each node of the evaluation tree is calculated as combination of the values of its successor nodes. Each end node is represented by a unique function, which might be a membership function based on fuzzy set theory [Zadeh, 65] for the case of a fuzzy criterion. Thus it is possible to take the desired amount of criteria and the relevant priorities into account.

5.4.2 Planning Supervision

Any replanning process is co-ordinated by the planning supervisor. The knowledge for scheduling the single processes and the crew-interaction to reach the goal-state can be represented as rule-base. Therefore, this process has been implemented by automata. For performing the selected actions scripts are used. Recommendations and requests to the pilot crew are also part of these scripts.

5.4.3 Information Acquisition

The information content of CASSY is represented as static and dynamic data base. As static data base a commercial Navigational Data Base (NDB) by Jeppesen is used, which is managed by a special process allowing quick and direct access to the required data.The dynamic data base consists of aircraft, crew and environmental data and is adapted to any event, which might change the global situation. Those events are reported to the information pick-up process. It is responsible for modifying the data pool, accordingly. Additionally, the events are structured in event classes. These classes are transferred to any CASSY core module, which has to react on the new situation.

5.4.4 Airport Evaluation and Selection

The airport selection module is able to evaluate airports for two different types of tasks

- determination of a rank order of alternate airports, for the case that a landing at the original destination is impossible
- selection of the next best emergency field for the case of an emergency, which requires an immediate landing.

The emergency field is displayed permanently on the CASSY graphic display to permit direct access in time critical emergencies. The display is updated about twice a minute. Alternates are evaluated on request and displayed with the major decision criteria in ranking sequence. Both emergency field and alternates are selected by the same process. Only the active evaluation diagrams are switched. The evaluation diagram for the emergency field selection exceptionally consists of safety critical criteria with the distance to the airport as one major point, whereas the alternate evaluation considers passenger comfort, service facilities and other safety uncritical criteria, too.

5.4.5 Trajectory Planning

Trajectory planning is a procedure of several stages. The minimum information needed is the starting location and the destination as well as knowledge about the global situation. More inputs about constraint checkpoints or altitudes specifying the trajectory can also be considered and accelerate the planning process.

First an altitude area is generated to ensure that the selected route copes with altitude constraints. Next step is route planning on standard as well as non-standard routes. The generated alternatives are presented visually to the cockpit crew for selection purposes. When a route has been selected - possibly after a few modifications - it is activated. During a radar vectoring phase, the radar vectoring module takes over these last steps for the estimated vectoring phase up to the expected reentry point into the flight plan.The profile is planned in detail with regard to airspace restrictions and ATC constraints. Afterwards it is being optimized for a given cost index and becomes the valid flight plan.These different stages of the entire trajectory planning procedure are illustrated in figure 9 and briefly described in the following paragraphs.

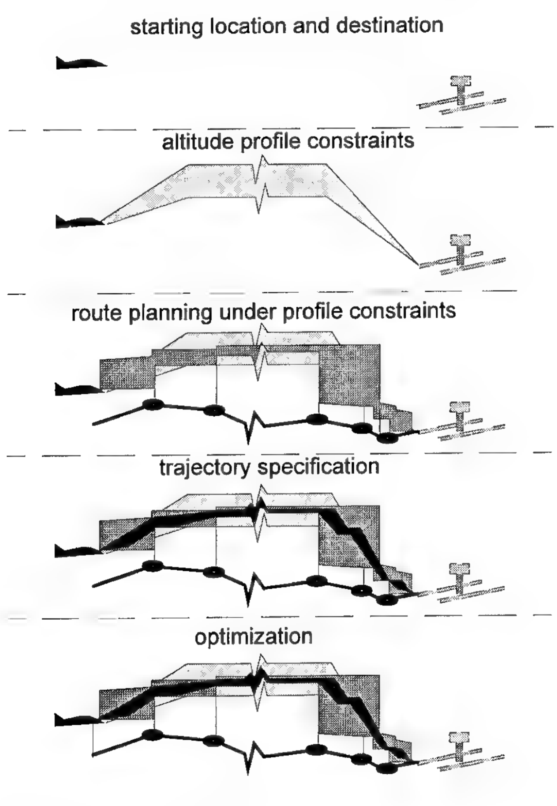


Figure 9. Trajectory Planning

5.4.5.1 Trajectory Specification

The submodule for trajectory specification is responsible for performing two different tasks. First of all altitude profile constraints have to be generated on the basis of knowledge about the actual position and the destination. The result consists of minimum and maximum cruise level, climb and descent rate, represented in the vertical plane. It is an important input for the route planning module to select an appropriate route, which enables carrying out the flight within the profile constraints. The calculation depends on a given cost-index, indicating the weighting of time and fuel for the flight. Climb, cruise and descent performance are taken from the published aircraft handbook that can be accessed as dynamic data base. This data base is adapted to atmospherical and aircraft conditions by a special performance monitor. The second specification stage starts after route planning. In this stage the selected route is matched with specific airspace restrictions. Points of intersections are figured out and the trajectory specificaton is modified to cope with these constraints. Thus, it is assured that any solution enhancement within these boundaries is executable with respect to airspace restrictions.

5.4.5.2 Route Planning

The inputs for the route planning module are at least the actual position, the destination, the generated altitude profile constraints and the planning mode. All of these inputs can be fed in by the crew, are replanning orders of the planning supervisor or a mixture of both of them. The route planning module is able to generate routes on different modes in sequence or in parallel, which depends on computer architecture. This is achieved again by switching evaluation diagrams and search strategy. The following search strategies have been applied with respect to typical AI searching algorithms [Nilsson, 82],[Boy, 91]:

- a simple best-first search for very time critical planning tasks
- a modified A* algorithm for rerouting in very large airspace areas
- a special algorithm developed for typical replanning tasks of CASSY, which is the normal AFP search strategy.

First, the airspace area is selected and the navigational data, such as waypoints and standard routes are drawn from the Navigational Data Base. The extension of the search space depends on the actual fuel situation. Then one of the above mentioned search strategies is applied in order to find the best flight path.The evaluation concentrates on single legs or standard route sections and uses the hierarchical evaluation principle explained in chapter 5.4.1. Some typical rerouting modes, represented in the evaluation diagrams are:

- standard routing:
  the route should content only standard route sections, as far as possible
- short distance - radio navigation assured:
  the route should be as short as possible, while selecting the best suited radio aids as checkpoints
- combination:
  standard routing, when not diverting too much from the direct routing, then include direct legs.

The principle of the AFP search strategy is to evaluate an initial leg, starting with the direct connection of origin and destination, and to try to improve the evaluation by including new waypoints. Any resulting path, which does not exceed the score of the initial leg, is closed. Any other is written on a waiting list in ranking order.

When all possibilities have been evaluated, the initial path is closed and the first path of the waiting list is opened. This procedure continues until a selected number of closed alternatives have reached a minimum score, or the waiting list is empty. Additionally, the procedure can be terminated at any time, since each evaluated route represents a continuous path from the origin to the destination. This enables the real-time application of the algorithm.

Due to the permanent and rapid increase in computing power, it will be possible in the near future to do without termination of the algorithm on a suboptimal path.

5.4.5.3 Radar Vectoring Estimation

It is an ordinary procedure that ATC takes the aircraft away from standard routing and guides it by radar vectors. When standard routing will be re-entered is uncertain. The reasons for this are separation purposes and the avoidance of unnecessary diversions. For providing all functions of the assistant system a continuous flight plan is required, though. Therefore, the AFP estimates the elongation of vectoring flight segment and the corresponding reentry point into the given flight plan.

The radar vectoring module keeps a knowledge base of typical ATC procedures, e. g. continous descent approaches and intercept procedures. All possible reentry legs of the flight plan as well as other standard routes within the interesting area are evaluated for coping with these procedures and other restrictions, such as minimum and maximum altitudes or speeds. The alternative with the best score is selected and inserted into the flight plan.

Each new ATC instruction causes an update of the estimation, which is also performed, when the estimation seems to be obviously wrong, which can be recognized by the conflict detection module.

5.4.5.4 Profile Planning and Optimization

When the profile planning module is activated, the route has been selected and the trajectory specified. The complete trajectory can be generated with regard to ATC-constraints. Some flight phases, especially take off and final approach are fixed by specific procedures and optimization is hardly possible. In addition, the ATC instructions have to be taken into account for the profile. The determination of the remaining trajectory, i.e. the part, not yet being specified, is restricted to

- selection of the appropriate flight level with regard to IFR procedures and ATC clearances
- selection of climb, cruise and descent velocities.

Keeping in mind, that the trajectory has to be flown by the pilot, the autopilot or a flight management system, very good results could be achieved by profile planning on the basis of the published aircraft performance. The application of an optimization technique would also be possible, but takes much more calculation time, which has been figured out by some tests. So, in real time the performance-oriented trajectory planning is applied in the AFP.

The profile planning enables a complete 3-D guidance of the aircraft and a partially realized 4-D guidance. The full 4-D planning will be the next development step.

5.4.6 Flight Plan Updating

The generated flight plan represents the final result of each planning procedure. It consists of primary flight parameters, i.e. altitudes, headings and speeds, as well as of secondary parameters, such as estimated times of overflight, altitude constraints and radio aids. The normative primary parameters: indicated air speed, magnetic heading, altitude and vertical velocity are displayed permanently within the respective flight instrument. The other parameters are indicated in the graphic CASSY display or can be requested by the pilot crew.

The flight plan is updated, when a new plan has been issued by other AFP modules, when a checkpoint has been passed and in time-descrete intervals. The first reason requires the generation of a completely new flight plan, whereas otherwise only times and vertical velocities are updated.

5.5 Piloting Expert (PE)

The PE essentially consists of a comprehensive knowledge base about the pilot behaviour when flying under instrument flight rules. Modelling of pilot behaviour within the PE is intended to be done in two ways, by a normative and an individual model: The normative model describes deterministic pilot behaviour as documented in pilot handbooks and air traffic regulations. Modelling is done primarily within the domain of rule-based behaviour, but covers admissible tolerances also. The individual model contains behavioural parameters of the individual pilot and is developed as a real-time adaptive component. In the following, the paper will focus on knowledge domain, representation and analysis of the normative behaviour model.

5.5.1 Knowledge Domain

Pilot behaviour in plan execution can be broken down into both certain aspects of situation assessment and action management components. Some special support functions are also part of the knowledge base. Behaviour modelling is done for all flight segments (taxi, takeoff, departure, cruise, approach, landing) and concerns the following tasks:

a) modelling of certain aspects of situation assessment:
   - recognition of actual flight segment
   - recognition of progress of plan execution corresponding to flight plan and procedures

b) modelling of pilot performance in action management:
   - primary flight guidance

(altitude, course, airspeed, power setting, climb/descent rate, pitch attitude)
- operation of flaps, landing gear, speed brakes
- radio navigation
- communication with air traffic control

c) model-based support functions:
- callouts (of important checkpoints, e.g. altitudes)
- checklist processing (normal, abnormal, emergency)

5.5.2 Analysis of Knowledge Base

To choose an adequate modelling formalism, the pilot tasks were analysed with regard to causal, temporal and structural relations. This analysis gave the following characteristics:

- Piloting tasks are strongly concurrent. This can be stated in the domain of situation assessment as well as in the parallel processing of several tasks (e.g. maintaining altitude, reducing airspeed, radial tracking, ATC communication).
- Processing of pilot tasks (e.g. radio navigation) is driven by situation-dependent choices of different rule domains (e.g. cruise navigation or approach navigation), this is a choice between (excluding) alternatives.
- The basic element within the considered tasks is always a causal relation, which can be formulated by a production rule (if ..., then ...).
- The situation space as well as the pilot's action space can by described by discrete states (e.g. flight segments, flaps setting) and state transitions (e.g. flight segment transition, flaps setting transition).
- State transitions are driven by discrete events (e.g. "passing station X", "reaching altitude Y", "system Z breakdown").
- Pilot behaviour can be broken down into several levels of abstraction, like flight segments and their decomposition into sub-segments in the domain of situation assessment as well as a holding procedure and its decomposition into single actions in the domain of pilot actions.

5.5.3 Representation of Knowledge Base

One of the most important objectives of this modelling activity was to obtain a homogenous representation of the considered pilot behaviour. Homogenity should be required with respect to low expense for software tools and - if available - to enable formal analysis methods. It is obvious that the knowledge representation method to be chosen must be adequate to the system characteristics named above.

In former systems knowledge was often represented by production rules and socalled production systems. However, production systems become difficult to control if the number of rules increases. Reasons must be seen in the lack of methods for structuring and decomposition. Finally, concurrency cannot be represented explicitly by production rules. Thus, modelling of pilot behaviour solely by production systems is no longer adequate.

Another alternative was the use of finite automata. However, the number of states which had to be modelled explicitly is enormous in view of the concurrencies to be considered.

These considerations led to the choice of petri nets by making use of different petri net classes, adequately matched to the particular properties of the knowledge domains considered:

- A considerable part of the knowledge is well representable by condition/event-nets.
- Another part of knowledge, even for modelling of multiple resources, requires at least use of place/transition-nets.
- Finally, a further part can, in principle, also be formulized by place/transition-nets, however for multiple identical model structures individual tokens are demanded and thus the application of high-level nets is suggested.

To bound the model complexity and the expense for net tools (primarily of the real-time tools) at first the class of place/transition-nets was chosen and used extensively for modelling [Reisig, 91].

Recently several petri net applications in the domains of civil aviation and aerospace arised, e.g. [Huck, 91],[Kreher etc., 93],[Lloret etc., 92]. Modelling is done for simulation as well as for analysis purposes, partly by high-level nets. Regarding the criticality of aviation/aerospace software and with respect to safety and the resulting certification processes, further - even industrial - applications should be expected in future.

5.5.4 Model Design Process

When applying petri net theory to a concrete technical process the problem encounters of missing general rules concerning the formulation of application knowledge into petri net constructs. In general, the question is: How does the transformation *real world → model* look like and which rules have to be applied ?

Typical questions arising in the modelling process are:

- Which real world components have to / may / must not be formulated as places / transitions?
- Which levels of net modularization are suitable?
- How can local testability of a large net construct be secured?

In the following, some characteristics of our petri net application are summarized. Especially we try to illustrate the design process, beginning with single production rules and rounding up with a hierarchically structured net system.

5.5.4.1 Semantic of Net Primitives

<u>Places</u>

Discrete states in the field of situation assessment and during pilot action procedures are represented by places (C/E-nets). Examples are flight segments ("final approach"), conditions for subsequent actions ("turn right after passing altitude A") and states of discrete aircraft systems ("flaps 20 degrees"). Multiple resources, e.g. redundant navigation devices, are represented by multiple marked places (S/T-nets). Within the scope of modelling pilot workload, limited pilot resources are also modelled by multiply marked places.

<u>Transitions</u>

Transitions are used to represent situation state transitions, e.g. between flight segments ("final approach → landing")

and discrete aircraft systems ("landing gear up → down"). In the domain of pilot actions transitions represent for instance changes between basic tasks ("cruise → descent"), navigation instrument settings and callouts of checklist items ("landing gear down ?").

Because transitions are typically used to model state transitions, their firing time is assumed to be zero. In case of a model-relevant state transition time the transition is decomposed into a place and timeless firing transitions.

5.5.4.2 Knowledge Transfer Production Rules → Net Construct

The knowledge base to start with consisted of production rules which had to be transformed into net constructs. In the following example the transfer of two simple rules is shown. The rules are:

- IF (flight segment = "Final Approach") AND (altitude << 50 feet over ground) THEN new flight segment: "Landing"
- IF (flight segment = "Final Approach") AND (recognized crew intent = "Missed Approach") THEN new flight segment: "Missed Approach"

Either rule can be represented as place-transition-place construct. The transitions are attributed by external conditions (altitude and crew intent criteria) (see fig. 10a/10b). It is evident that the identical net pre-conditions of both transitions ("Final Approach") lead to a joined net construct (see fig. 10b). Analoguously, if pre- and post-states of different rules are identical, they are connected sequentially.

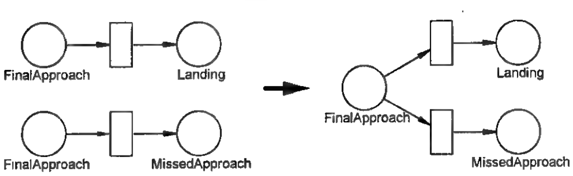


Figure 10a Figure 10b

5.5.4.3 Modular Construction

Size and complexity of the knowledge to be represented by petri nets for the application domain considered require extensive use of modularization techniques. Several requirements have to be satisfied:

1) Subsystems must be testable and analyzable on the local level. For this reason, activation and deactivation of coupling mechanisms is needed.
2) Because no tokens may be inserted or removed dynamically, all subsystems have to be strongly connected and marked.
3) Token flow between subsystems is not allowed. Nevertheless, implicit token flow is realized by the activation of subsystems (see 5.5.4.4).
4) Requirement 3) implies that access to places of other subsystems is permitted read-only.

The literature names different kinds of modular net construction, e.g. place and transition refinements, place and transition fusion sets, invocation transitions etc. [Vogler, 92],[Huber etc., 90]. According to the mentioned requirements we choose the place and transition fusion sets for our application. This method allows to give a place or a transition multiple graphical representations, even in different nets.

Modular construction often requires access to state or state transition information established in other subsystems. This information must be accessed in a read-only way. Thus we use two coupling mechanisms, both construed of place and transition fusion sets.

- read-only access to state
  The required state information is imported into the client net by place fusion. The place can now be accessed by test (double) arcs. Thus no token flow between the nets is allowed (see fig.11a).

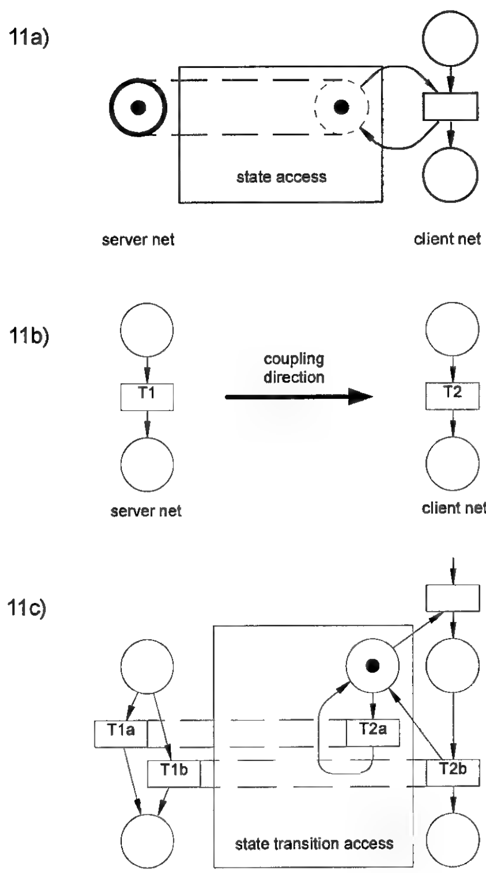


Figure 11. Coupling mechanisms

- read-only access to state transition
  This coupling mechanism is used to model an unidirectional dependence between two transitions (see fig.11b): Firing of T1 should be a precondition for firing of T2, but T1 should fire independently.
  The construct is realized by splitting transition T1 into T1a and T1b and by importing them into the client net via transition fusion (T2a, T2b). A place complement is

needed to guarantee firing of either T1a or T1b. Fig.cshows that the state transition of the server net (firing of T1a or T1b) is not restricted by the state of the client net, while firing of T2b is coupled with the state transition of the server net (firing of T1b). Such coupling ist often applied for reset purposes.

5.5.4.4 Hierarchical Construction

The design of the net model is done in a top-down way. In many cases already modelled behaviour aspects have to be expanded by a more detailed model. Of course, net models cannot be extended boundless (graphical representation, testability etc.). Thus it has to be decided which parts of the net model are suited to be located in a subsystem and which coupling mechanisms are applied. In many cases it is desired to refine a state which is represented in the coarse net by a single place. A direct replacement of the coarse state by a subsystem does not comply with the modularization requirements mentioned above (primarily 1)). Since the coarse state carries semantical information (e.g. accessed by other nets), it is essential not to substitute the coarse state (as done by place refinement). For these reasons we refine states by duplicating their "interface" transitions into a subsystem, where the coarse state is modelled in more detail. In case the coupling mechanisms are deactivated, this construction preserves the behavioural properties of the coarse net. To fulfil requirements 1) and 2), a marked complement place is added to the subnet (see fig. 12).

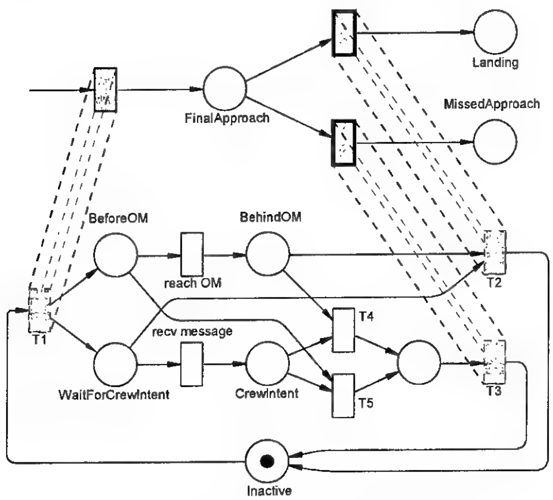


Figure 12. Refinement of Place "Final Approach"

Coupling of the two nets is done by transition fusion sets of the interface transitions. This construction is an extention of pure place refinement (see [Vogler, 92]).

Figure 12 illustrates this technique using the example of figure10: We consider the place "FinalApproach", i.e. the decision between "Landing" and "MissedApproach" has to be modelled in more detail.

The upper path of the subnet in figure 12 represents a lateral decomposition of the flight segment "Final Approach". This is done to pay regard to the outer marker (OM) beacon. A recognized crew intent "Missed Approach" is told to the net by a message from the CASSY-module PIER. This message has to be received concurrently to the described flight segment decomposition. This is modelled by the places "WaitForCrewIntent", "CrewIntent" and the transition "recv message" (lower path of the subnet). In case no crew intent message is received, the flight segment "FinalApproach - BehindOM" terminates under normal conditions by firing of transition T2 (altitude condition, see section 5.5.4.2, rule 1). In case a crew intent occurs, the subnet terminates via transition T3. Different actions are performed dependent on the actual flight segment (transitions T4, T5).

5.5.4.5 Process Interface

Modelling of pilot behaviour in plan execution can be separated into situation assessment and action components (see 5.5.1). As a basis for a rule-based situation assessment model a discrete situation space with well-defined state transitions has to be established.

The rule-based situation assessment process is characterized by a permanent consideration of all possible state transitions with regard to the actual situation state vector. These state transitions are typically defined as discrete limits within the - more or less - continuous state space of aircraft and environment (e.g. "passing station X", "reaching altitude Y").

For the assessment of the actual situation the state transition itself suffices. Nevertheless, the processes leading to this state transition influence the dynamics of situation assessment. Obvious questions like "what is earlier reached - station X or altitude Y ?" show that the causal structure of the underlying processes have essential effects on the assessment results.

For an overall investigation of the dynamics of situation assessment, the causal structure of aircraft and environment has to be made accessible to analysis methods. This means these systems have to be modelled by petri nets, including qualities like "x happens before y".

For real-time situation assessment state transitions within the net model have to be executed dependent on external (aircraft / environment) conditions, in the following called "firing conditions". These firing conditions can be understood as states within a - not realized - aircraft / environment net model (see fig.13).

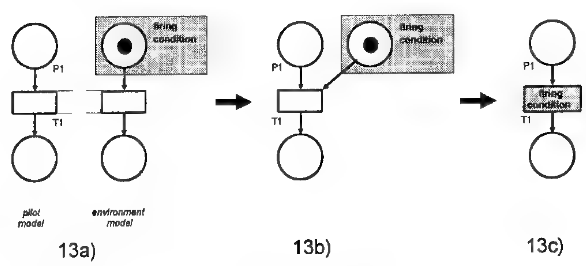


Figure 13. Representation of firing conditions

These two net models can be connected by a common transition. The marking of the places P1 and 'firing condi-

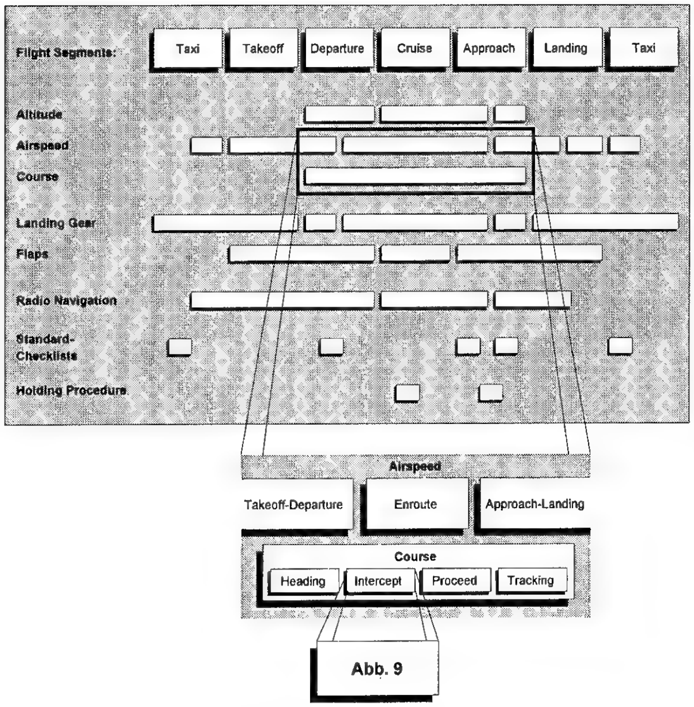


Figure 14. Model Structure

tion' enables the firing of transition T1. With regard to the firing of T1 the net structure containing 'firing condition' can be neglected. This leads to a compressed representation, see figure 13b. A disadvantage of this representation is that a token has to be inserted when the condition occurs. Besides, if the condition is left without having fired T1, the token has to be removed to avoid subsequent, faulty firing. For this reason we formulate this firing condition briefly as transition attribute (guard), see figure 13c. A back-transformation to the other representations, e.g. for analysis purposes, can easily be done.

5.5.5 Model Structure

Figure14 gives an overview of the (strongly simplified) model structure. The structurization of the net model was done according to knowledge structures as far as possible.

The primary rationale of structurization are the pilot tasks within plan execution: recognition of flight segment, primary flight guidance (altitude, course, airspeed), system operation (gear, flaps, radio navigation) etc., see section 5.5.1 for details. These are typical examples for concurrent tasks.

To come up with subnets of handy size and complexity (not more than 10-15 places, reset constructs excluded), most of the tasks need further subdivision. For this purpose subclasses of behaviour within the main tasks had to be identified. An efficient structurization was done by separating behaviour with regard to different situation characteristics. The resulting behaviour classes are always related to excluding situation elements, therefore they are exclusive alternatives. The situation characteristics can mainly be attached to two groups: flight segment subsets (e.g. departure airspeed behaviour) and orders derived from the flight plan (e.g. course behaviour for "proceed to station X").

Figure 14 shows the concurrent task models in vertical direction. Alternative (excluding) submodels related to flight segments are drawn in horizontal direction. Submodels are represented by white boxes; their size, complexity and hierarchical depth (i.e. number of subnets) differs widely. Nets for coordination purposes and their couplings with task model nets are also neglected in this illustration.

A small part of the model structure is zoomed out and discussed in more detail. Figure 14 shows the course model and a part of the airspeed submodels. Subdivision of airspeed behaviour is done with respect to to flight segments (takeoff-departure, enroute, approach-landing subnets). The actual course selection behaviour class is derived from the flight plan and is a choice between "turn to heading H", "intercept course C of station S", "proceed to station S", and "tracking from station A to station B". A simplified "Intercept" subnet is described in the following (see fig. 15 / 16).

<u>Example</u>

An interception is carried out to reach a given (magnetic) course to a given station (e.g. a radio navaid). This can be required within published departure or approach procedures or can be ordered by air traffic control. In the general case, an interception covers 4 sections (see fig.15): turning to a special intercept heading (S1), maintaining on intercept heading (S2), turning to given course (S3), tracking on given course (S4). Sections are skipped if the aircraft fulfils the characteristics of a following section, e.g. if the aircraft is already on intercept heading at the time the procedure is started, section S1 is skipped.

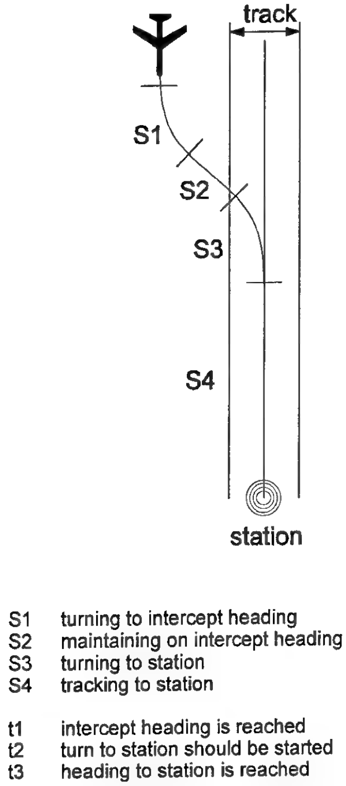


Figure 15. Intercept Procedure

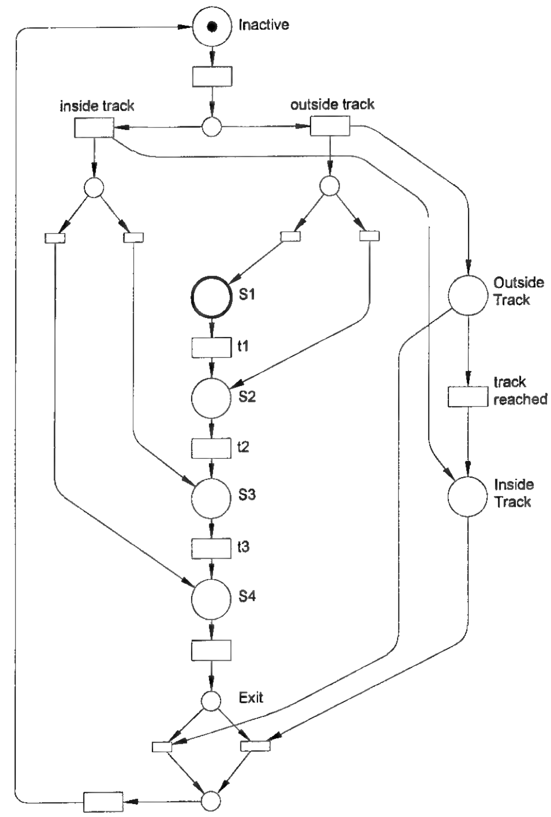


Figure 16. Intercept Net

In addition to this heading behaviour, the given course to the station and an admitted course deviation specify a lateral track. After having reached this track, the aircraft should not leave the track until a new lateral procedure is started (see fig. 15). In this example it is assumed that the transition point between section S2 and S3 (start of second turn) is always placed outside the track.

The intercept net is mainly characterized by two concurrent constructs:

- A sequence of 4 places ("S1" to "S4") represents the 4 subsequent behaviour sections described above. The transitions connecting these states are attributed with heading conditions or other lateral conditions.
- The places "OutsideTrack", "InsideTrack" and the transition "track reached" are used for the modelling of the tracking performance mentioned.

After the net became active several conditions have to be considered within the initial section of the intercept procedure (not discussed here in detail). A further concurrent construct is needed to enable a net reset from all (stable) states, for instance in case of a changed flight plan (not shown in fig. 16).

As final result of the modelling process we expect at least 250 subnets with about 2500 places and 4000 transitions. At the time being, the model covers 1800 places and 2800 transitions in 170 subnets.

In the following, the size of this effort is summarized. This petri net activity was started at the end of 1991. Net modelling was done by the authors and in part by engineer students (5 persons, together 22 man-months). A part of the students had knowledge within the application domain (e.g. helicopter pilots). The students did the net modelling of small knowledge domains with great interest. The tools were mainly developed by computer science students (5 persons, together 24 man-months).

5.5.6 Analysis - Goals and Results

When the development was started, model testing was done mainly by simulation. Only few simple properties like connectedness were checked automatically after parsing the net declaration. Simulation tests have to be carried out anyway, even to check numerical results and the interfacing of the pilot model. However, the main problem of testing only by simulation runs is that it is impossible to reach all (critical) net states and state transitions within the test run. With respect to reversibility all net states are critical, because if a reset condition occurs and the net is unable to perform the reset successfully, this must lead to deadlock or at least to malfunction of the net and the parent nets.

For this reason formal analysis methods are applied to the net model in the meantime. The analysis strategy is "bottom-up", thus in a first step all subnets are checked to satisfy some obligatory qualities. These are at least:

- strong connectedness
- boundedness
- reversibility
- liveness and safeness

Analysis was done for all 170 subnets by the analysis tool

INA [Starke, 93],[Starke, 90]. Net analysis proved about one essential defect in every tenth net, mainly incomplete reset structures (remark: generally successful simulation runs are done since one year !).

The next analysis step is to combine (already checked) subnets into more complex net systems, step by step, and to prove the required properties again for the more complex net. This has not been done yet.

Besides these general properties there are other special properties which can be derived from the pilot model specification. Examples are:

- exclusive states
  (e.g. exclusive flight segments, exclusive aircraft system states)
- state sequences
  "states S1, S2, ..., Sn have to occur / may never occur subsequently"
  (e.g. flight segments)
- state refinements
  "activity of refined state Sr requires activity of coarse state Sc"
- predetermined reset procedures
  "firing of transition Tr carries over every net marking to the initial marking"
  (e.g. subnet reset)

These properties can be proved using facts, invariants or special evaluations of the reachability graph (critical with large models). Some of the listed properties have already been investigated on the basis of the reachability graph generated by INA.

For extensive verification a checking tool enabling the formulation of logical terms (numerous and/or-operations, negations) and avoiding the calculation of the reachability graph is desirable.

5.5.7 Tools

As a main requirement, the net model, as presented, is to be used not only in the design phase but also as final implementation of the CASSY-module Piloting Expert. For this purpose, a real-time petri net interpreter was needed. This central role of highly integrated real-time net interpretation is an atypical aspect of this application and has some unfavourable effects on the suitability of commercial petri net tools. In the following, the main tool requirements are summarized:

- availability of real-time tools (interpreter and monitor) on the CASSY hardware platform (Silicon Graphics workstation)
- interpreter interface to program language C for integration of transition guard / action functions (process interface)
- strict separation of interpreter (simulator) and graphical user interface
- graphical and textual net declaration (especially important to large nets and declaration of transition attributes)

Because of these requirements only a fraction of the commercial tools could be applied.

A description language for place/transition-nets was defined enabling the declaration of modular constructs and process interfacing by transition attributes (guard and action functions).

By use of the commercial tool Design/OA [Meta Software Corporation, 91] a graphical editor was developed which supports the required coupling and refinement methods and the local treatment of subsystems.

The net interpreter is integrated in the real-time data processing of the assistant system and does not have any graphical interfaces. The central requirement for interpreter development was to gain response times not dependent on net size and nearly independent on the number of actual active transitions. The process couplings are achieved by use of an open interface to progamming language C. Transition guard and action functions implemented in C are automatically linked to the net data structures and to the interpreter kernel.

Debugging of net simulations is supported by a graphical monitor system using OSF/Motif. The monitor receives actual marking information from the interpreter. Transition firing (overwriting of transition guard functions) can interactively be done by the monitor. Special attention was paid to a net activity dependent choice of presented information. Because of system size this information reduction is indispensable.

As presented in the last section, the commercial tool INA is used for net analysis.

5.6 Dialogue Manager

While the interface between CASSY and aircraft is realized using data bus systems, the task of the module Dialogue Manager (DM) is, as mentioned before, to control the information flow between CASSY and the crew in either direction. This interface is established using speech recognition, speech synthesis and a graphic colour display for complex information output (see fig. 17). The DM controls the output information flow as well as the language model to define possible speech input.

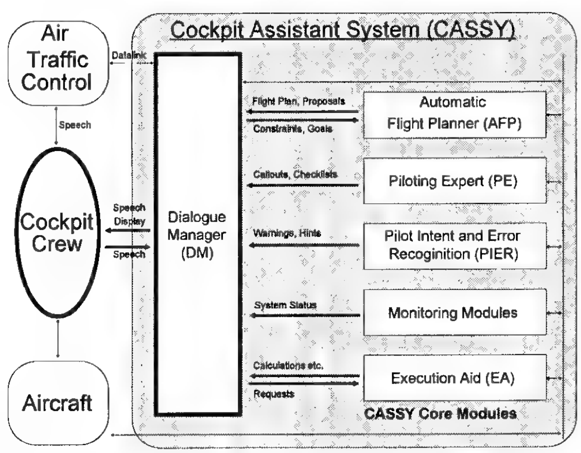


Figure 17. Structure of Dialogue Manager

As long as a data link is not available, CASSY is picking up the ATC instructions via speech recognition. Thereby the

regular pilot procedure of obligatory acknowledgement of ATC instructions is exploited.

5.6.1 Output Information Flow

In CASSY a graphic colour display and a speech synthesizer are used as output devices.

The audible perception channel is independent from the pilot's visual fixation and is therefore suitable to be used e.g. for warnings and hints since in some flight phases like final approach the visual perception channel is highly demanded by instrument cross checks or other visual tasks. The used phraseology for speech output is based on the rules of radio communication [BFS, 89] and on current crew coordination concepts which are defined to ensure an error free information exchange between pilots and air traffic controllers and within aircraft crews. Different catagories of output messages are referenced by different voices.

If the message is too complex to be presented by speech output, the graphic colour display is used. The display can be used in several modes (e.g. radio navigation chart in heading or noth up mode, alphanumeric board, airport approach chart etc.) which ever seems appropriate for the given output. The perception of some complex verbal messages is graphically supported on the display while at the same time the message is issued from the speech synthesizer. If the message refers to numerical values, the message is printed on a reserved area on the display. If the message refers to a geographic location the respective area is marked on the display.

The CASSY output information, represented as 130 different messages, comprise

- warnings and hints to avoid or correct an observed deviation from the flight plan,
- additional information to support flight plan execution like data base requests for frequencies of navigation aids,
- checklist and briefing items,
- the current flight plan or flight plan proposals including the describing information (permanently depicted on a radio navigation chart on the display).

Warnings refer to deviations from the flight path, to aircraft systems like instrument or flaps and gear settings and to flight phase relevant tasks. Depending on the importance in the actual situation the text is chosen to emphasize the message (e.g. "check heading", "turn left heading 2 2 0"). Warnings are issued in two different voices depending on the severeness of the observed error.

The services of the EA, i.e. results of data base or calculation requests, are presented by speech output supported by alphanumeric messages on the graphic display. Checklists and briefings are sequential tasks and the respective messages follow each other directly. Whereas checklist items are usually simple and can be presented verbally, briefing messages refer to geographic flight plan check points and are supported by marks on the appropriate presentation of the flight plan (airport departure/approach chart) in the graphic display.

As the CASSY core modules provide information for messages independently of each other it may happen that there are some messages to be transferred to the crew at the same time. Therefore, the messages are stored before they are transmitted to the output devices in order to determine a suitable message output order. The messages are evaluated and the one with the highest score is chosen. The order is influenced mainly by:

- the urgency or appropriateness of the message in the given situation
- the thematic relationship to last issued message

If the message is a warning, the degree of violation of respective safety tolerances determines the message urgency. The appropriateness of other messages are functions of the current flight phase. For example, briefing messages for a missed approach procedure are more appropriate when the missed approach occurs than in other flight phases.

In sequences like briefings or checklists the thematic relationship to the last message is very important. The respective messages are issued one after the other so that the pilots' sequential task can be completed. This order should only be interrupted by very urgent messages.

Each message interrupts the output information flow for a certain time in order to avoid a flow of messages. The duration of the pause depends on the task associated with the issued message.

The knowledge about the messages are declaratively represented in frames.

5.6.2 Input Information Flow

Speech recognition is chosen as input device. Speech input is suitable for discrete infomation transfer. In aircraft cockpits it is appropriate to avoid long head down times to input information via conventional input means [Taylor, 92].

The input information can be broken down into

- crew inputs
  - flight planning commands,
  - CASSY configuration commands, e.g. display configuration,
  - data base requests and performance calculations,
  - aircraft system configuration commands using CASSY's access on aircraft systems,
- air traffic control instructions (ATC)
  - traffic guidance commands.

As long as the planned data link for communication between air traffic control (ATC) and aircraft is not available ATC instructions must be fed into the system by the crew. This is achieved by picking up the communication between ATC and the pilot. As the pilot is obliged to give his radio call sign (i.e. usually the flight number or registration number of the aircraft) before or immediately after reading back the controller's command, the recognition of the call sign can be used to discriminate a crew command from an ATC command. A wordspotting algorithm is used to detect the radio call sign within the speech input of the pilot.

As for speech output, the language model for speech recognition is based on aviation phraseology. While the speech output texts are chosen from given guidelines for communications the language model for speech input is additionally

based on experiences of actual communication between pilots and ATC .

Several applications of speech recognition as input show that the amount of possible combinations of input commands may lead to very complex language models. Often, the complexity of the used language model is the reason why speech recognizers might show poor recognition rates and are therefore not considered as satisfying input devices. Even if the expected speech is reduced to command structures like "turn left to heading 1 8 0" the many different possibilities of parameters like 'heading' (i.e. 360) lead to an overall language model producing billions of possible sentences. In this application a basic language model which is not reduced by the knowledge of the actual situation leads to a syntax which can be described by a perplexity of 16 and a vocabulary of 218 words (excluding the names of geographic location and navigation aids). In order to reduce the complex language model, the current situation of crew and aircraft must be considered. Not only the amount of parameters can be reduced but also the amount of input commands themselves as some of them occur only in certain situations.

In this application the complexity of the language model is significantly reduced since knowledge about

- the flight phase, flight plan and expected crew actions,
- aircraft position and surrounding geographic area and aircraft data,
- actual information output to the crew and a classification into crew and ATC inputs

are taken into account.

The representation of expected crew actions, as modelled by the PE, is used to predict task relevant speech inputs of the crew (fig. 18). The state of each command which refers to autopilot settings, instrumentation and aircraft configuration settings or requests for support of required procedures like checklists, is evaluated. Each state is connected with a function that gives an evaluation of the probability of occurrence according to the pilot model.

Reactions of the crew to information output is considered estimating appropriate input commands. A warning which refers to a wrong altitude increases the probability of a respective autopilot command. This approach is also used to enable a DO-IT speech command as an answer to warnings like "descend to 5000 feet".

As aircraft systems settings can be manipulated by a speech command which is interpreted and carried out automatically by CASSY, the pilot may use commands which are closer to his mental representation (e.g. navigation beacons may be referred to by their names instead of difficult frequencies).

The reference numbers of the commands and their evaluations are stored in a list of currently possible commands. This list is the basis for both the complete, context depending language model and the assignment of the reference number to a speech input of the crew. Input commands which occur only in certain, time constrained situations (e.g. planning or checklists), are added to the list of probable commands when these situations apply and dropped afterwards.

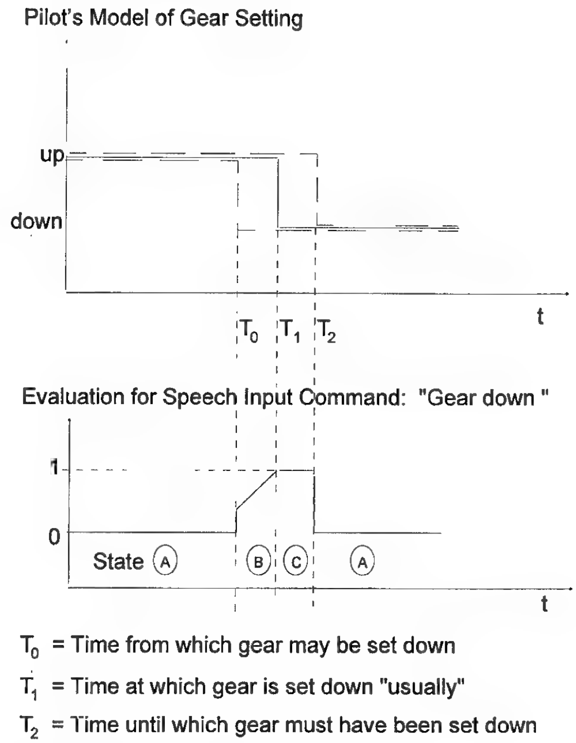


Figure 18. Evaluation of input commands

Each input command (103 for crew, 66 for ATC) is represented as a frame. Its semantic meaning is coded as a reference number plus describing parameters. The slots of the frame contain the actual possible parameters derived from respective decision trees, the vocabulary and syntactical constraints as transition network describing each way to speak the command and an evaluation representing the actual probability of occurence.

An equivalent approach to predict ATC instructions would require a similar model of the controller and the actual traffic situation which is not available in CASSY. Therefore, the description of the ATC language model is based only on the current flight phase and constrained to the actual geographic area.

The number of sentences for the crew language model varies between several thousands (perplexity 2 - 3, vocabulary 90 - 150 words) and for the ATC model between several ten thousands (perplexity 4 - 5, vocabulary 90 - 150 words).

The speech input is prompted on the CASSY display for verification. The recognized command is evaluated on its recognition score and on its actual relevance to find out whether it can be accepted automatically or whether it should be explicitly confirmed by the crew.

6. REVIEW OF SIMULATION EXPERIMENTS AND IN-FLIGHT FIELD TRIALS

The range and kind of provided functionalities as described above to some extent and the way, they are realized, result from extensive simulation experiments and consultations of

pilots at several decisive development step. After the successful completion of these simulator trials flight experiments were performed in June 1994 [Prévôt etc., 95].

6.1 Simulation Experiments

The first significant results have been gained in extensive simulation experiments with the CASSY predecessor system ASPIO (Assistant for Single Pilot IFR Operation) in 1990 at the German Armed Forces University in Munich [Dudek, 90]. This system had been prototyped primarily for the approach and landing phases of flights in the Munich area. The experimental environment consisted of a very heterogeneous hardware from Personal Computers to VAX-stations. For speech recognition a single word recognizer was used. With respect to this architecture the results were astonishingly promising. The design philosophy proved to be well accepted by the pilots. The significant improvement of flight accuracy and acceleration of planning and decision making tasks were further factors, which gave way for continuing projects and industrial support [Onken, 92].

The following activities were aimed at the two men cockpit of regional aircrafts. One major part of system improvement was extending the support functions for all flight portions. Integrating a dialogue management module was stressed as well as exploiting crew modeling techniques. Consistently, the pilot intent recognition became part of the system, too. After two more years in 1992 a first CASSY prototype could be demonstrated in the one-seat fixed base flight simulator at the university. In the meantime the computer hardware had been changed to a homogeneous workstation architecture and continuous speech recognition systems. Again, airline pilots were consulted for further improvements. The flight procedures were adapted to major airline procedures and specific details of the man-machine cooperation were discussed and improved.

In 1993 the system was tested in the DO 328 flight simulator at DASA, Friedrichshafen. Flights in the Frankfurt area have been simulated with different pilots. Good pilot acceptance was for the planning and the monitoring functions [Onken and Prevot, 94]. The speech input was marginally accepted because of technical shortcomings. Therefore, the speech input part of the Dialogue Manager has been improved and is still under improvement.

Before finally integrating the system into the test aircraft, flights have been simulated with the hardware in the loop in the simulation facilities of the German Aerospace Research Establishment (DLR) in Braunschweig. In these simulator runs the software was checked and the test pilots for the flight experiments were introduced to the functionalities of CASSY and the experimental environment.

6.2 In-Flight Experiments

The flight experiments were aimed at evaluating the CASSY performance in the real aviation environment. The system was integrated into the experimental cockpit of the Advanced Technologies Testing Aircraft System ATTAS of the DLR and typical regional flights in high traffic areas were performed. In the following the experimental environment and the flight scenaries are presented.

6.2.1 Experimental Environment

The flying simulator ATTAS, an especially developed modification of the 44-seat cummuter jet VFW 614, is equipped with an experimental fly-by-wire flight control system and a versatile computer and sensor system. Beyond many other test programs it is used as the airborne segment in DLR's air traffic management demonstration programme [Adam, et al, 93] and is equipped with very good facilities for testing complex on-board systems in instrument flight scenarios. In addition to the two safety pilots seated in the front cockpit, the ATTAS aircraft can be flown by the test pilot in an experimental cockpit, which is installed in the rear cabin directly behind the front cockpit. The experimental cockpit is a generic flight deck (one seat) with side-stick, airbus display and autopilot techniques and ARINC control panels. Therefore, it represents a realistic pilot working environment for IFR operation. The CASSY hard- and software has been integrated into the experimental cockpit.

The hardware of the assistant system, consisted of

- an off-the-s helf Silicon Graphics Indigo (R 4000) workstation to run the core modules of the assistant system connected to the ATTAS experimental system via ethernet
- a PC/QT equipped with a Marconi MR8 PC-cart providing speaker dependent continuous speech recognition with a speech button on the side stick
- a DECtalk speech synthesizer with various voices for speech output connected to the ATTAS intercommunication facilities
- a BARCO monitor (about 25cm) connected to the graphics channel of the SGI-Indigo and built into the experimental cockpit.

The computers were located in the rear of the main cabin. There are several experimentor work stations in the aircraft. One was equipped with a laptop for starting and maintaining CASSY.

6.2.2 Knowledge Acquisition

During the flight tests CASSY has been running throughout the complete flights from taxi-out to taxi-in. All data, which CASSY received via the avionics data bus, have been recorded with a frequency of 10 Hertz. All in- and output messages have also been recorded and every time, the flight plan had changed because of a major planning activity or when a checkpoint has been passed, the whole situation representation has been stored. These data enable a replay of all flights and a reproduction of all situations.

The presented results have been gained by observing the behaviour of the pilot and the intelligent assistant during the flights on-line, by off-line evaluating the collected data and in debriefings immediately after the flights. Two professional pilots served as experimental pilots and additional pilots from Lufthansa German Airlines were participating as observers.

6.2.3 Flight Scenarios

A total amount of about 10 flight hours has been performed, comprising eight flights from the regional airport Braunschweig (EDVE) to the international airports of Frankfurt (EDDF), Hamburg (EDDH) and Hannover (EDVV) at which a missed approach procedure was conducted before returning back to Braunschweig.

Table 1. Flight test scenarios

Flight	T/O T/D	time airb.	G/A in	after	ATC instr.	Pilot
1	EDVE	1:03	EDDH	0:33	26	1
2	EDDF	0:50	*inflight*	*simul.*	13	1
3	EDVE	1:27	EDDF	0:43	27	1
4	EDVE	0:50	EDVV	0:09	24	2
5	EDVE	1:32	EDDF	0:41	32	1
6	EDVE	0:57	EDDH	0:32	27	2
7	EDVE	0:57	EDDH	0:31	24	1
8	EDVE	0:58	EDDH	0:31	21	1
9	EDVE	1:31	EDVV	1:14	42	1

Flight no. 2 has been an in flight simulation of departure and approach to Frankfurt, which was necessary to investigate certain incidents, which would have been safety critical in the real Frankfurt area, e.g. descending below the minimum safe altitude. In all other flights nothing has been simulated and no special situations have been provoked, since the system should be evaluated in the real environment, which includes coping with all events, which occur during an IFR flight in a high density area.

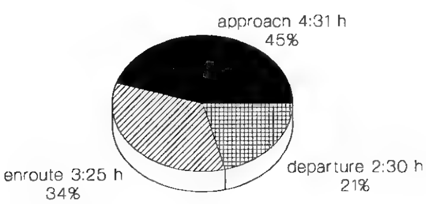


Figure 19. Distribution of flight phases

6.3 Experimental Results

One important result held true throughout the complete test program: There was no significant difference in system performance between the flight tests and the simulation trials. Consistently, the following discussion of results concentrate on the major questions concerning the real environment rather than system performance.

6.3.1 Operationality of the Interfaces

To evaluate the speech recognition performance three different speakers made the speech input during the flight tests, summarized in table 2.

Table 2. Speech Inputs

Speaker	Time	Inputs
Pilot 1	8:18	324
Pilot 2	0:50	36
Experimenter	0:57	56

In their first flight pilot 1 and pilot 2 were not very familiar with the speech recognition system and the specific syntax to be used. In flight no. 6 a CASSY experimenter made the complete speech input for the pilot. He was familiar with the syntax and the speech recognizer from simulation experiments. The results are shown with regard to recognition performance.

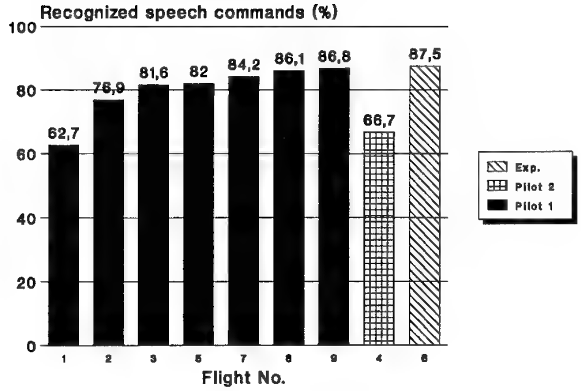


Figure 20. Percentage of recognized speech commands

Obviously, speech recognition inside the noisy aircraft is possible. It takes some time for the pilot to become familiar with the recognizer and the syntax to be used. This learning process can also be done in the simulator, as the flight with the experimenter has shown. The achieved percentage of recognized speech commands is almost of the same level as could be achieved in simulator runs with the same recognition system.

For entering the ATC commands into the system, two different experiments have been made throughout the flights. 92 ATC commands of a total of 236 have been fed into the system by the pilot using speech. The remaining 144 commands have been keyed into the system by one of the experimenters onboard the aircraft immediately after receiving the message, to simulate a data link from the ground into the aircraft. This took some seconds. The pilot reacted to the commands at the same moment he received the ATC message, but acknowledged the command with some time lag. This time lag resulted in delayed reaction of CASSY, which sometimes led to unnecessary warnings and hints. The percentage of occurence of these incidents compared to the respective number of ATC messages and the mean phase lag is illustrated in figure 21.

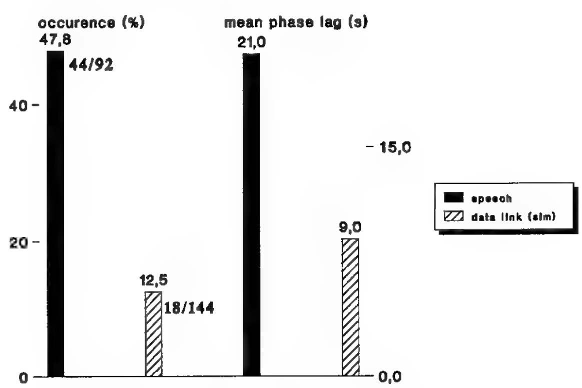


Figure 21. Unnecessary warnings or hints resulting from time lags in entering ATC commands

This effect was typical for the one-pilot configuration of the experimental cockpit. The figure illustrates the importance

of a fast and powerful ATC interface. Optimal system performance can only be achieved with a digital data link.

6.3.2 Situation Assessment with Respect to Pilot Behaviour

The basic requirements described in chapter 2.1 point out the necessity for a complete understanding of the global flight situation. To get an impression of the situation assessment capabilities the duration of discrepancies between the actual and the expected pilot behaviour has been related to the total flight time. This has been done on the basis of the stored data for the six flights 2, 3, 5, 6, 7 and 8.

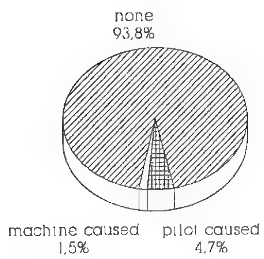


Figure 22. Discrepancies in actual and expected pilot behaviour

Figure 22 indicates that for almost 94 % of the flight time in a high density environment the pilot and the machine assessed the situation equally, because otherwise they would not expect or perform the same action patterns. A total amount of 100 incidents leading to warnings have been evaluated to find out the reasons for the warnings and the consequences they had. All incidents have been related to one of the three categories: pilot error, pilot intent and machine error (i.e CASSY errors in this case).

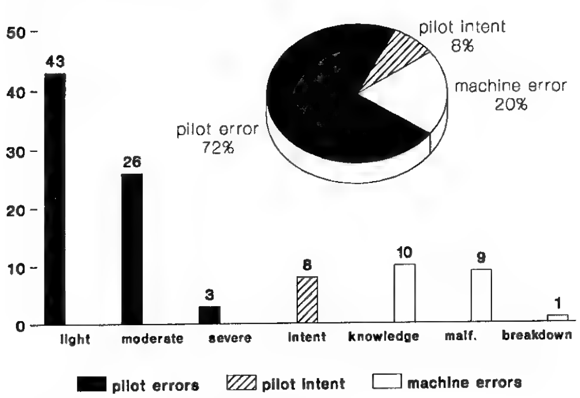


Figure 23. Error classification

In five cases of the intentional deviations from the flight plan the intention was autonomously figured out by the assistant system and the flight plan has been adapted, accordingly. In three cases the pilot had to inform CASSY about his intention.

Half of the machine errors were caused by an incomplete knowledge base, e. g. insufficient modeling of the aircraft performance and the other half by malfunctions of CASSY, i.e. software implementation errors due to less rigorous application of software development procedures. In one case such a malfunction led to a complete breakdown of the assistant system. In all machine error cases the pilot realized that a wrong warning was issued by CASSY. No negative influence on the pilot's situation assessment could be observed. In the one breakdown case, the complete CASSY system had to be restarted in flight, which took about 15 seconds. The only pilot input needed for such a recovery procedure is the flight destination. In all other machine error cases the warnings disappeared autonomously, when the incorrect assessed maneuver had been completed by the pilot.

Concerning the pilot errors the light errors are considered to result in an inaccurate or uneconomical, but safe maneuver. Moderate errors, probably would lead to a safety critical situation, and severe errors surely would lead to a dangerous safety hazard unless an immediate correction is made. All pilot errors, which occured during the flight tests, were detected by CASSY. All moderate and severe errors as well as about 70% of the light errors were immediately corrected by the pilot after having received the warning or hint.

6.3.3 Flight Planning and Decision Aiding

CASSY's flight planning capabilities have been stated by the experimental pilots and the observers as very impressive. As a matter of fact, all planning proposals have been accepted and none of the autonomous radar vectoring predictions has been modified or caused any doubt from the pilot. The time needed for planning a complete flight from one airport to the other is illustrated in figure 24.

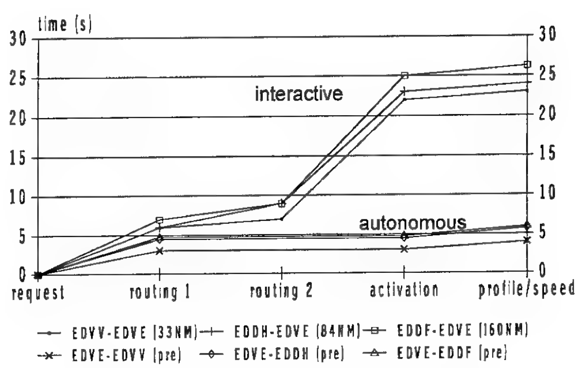


Figure 24. Duration of flight planning actvities

Before every flight the flight destination and the departure runway were entered into CASSY and an autonomous planning of the complete flight was initiated. After the go around procedure at this destination the pilot initiated an interactive planning to return to the airport, from which he had departed, by entering its name. CASSY elaborated and presented two routing proposals in parallel, which the pilot could select from or modify. After the selection the trajectory profile was planned in detail and recommended speeds, times of overflight, radio aids etc. were inserted.

The distance to the destination had only little impact on the duration of planning. The autonomous planning took between 4 and 6 seconds, the interactive replanning up to 26 seconds, of which the pilot needed about 16 seconds to decide for a proposal. This confirms the approach to replan autonomously, when the flight plan must be generated very fast. When there is more time available, replanning can be done interactively, too, in order to keep the pilot more involved.

6.3.4 Pilot Acceptance

The acceptance of the planning and monitoring functions of CASSY was at least as good as in the previous simulator trials [Onken and Prévôt, 94]. All pilots participating in the evaluation attested CASSY a nearly operational performance and a very promising concept. It was noted that the CASSY functionalities for enhancement of situation awareness, situatuation assessment and monitoring as well as the good planning capabilities are completely in line with human-centered design.

7. CONCLUSION

The time has come that future cockpit systems no longer will be designed on a vague basis of specifications. The advances in technology have brought about means to systematically reflect requirements for human-centered automation into clear-cut specifications and cockpit system development. Machine functions will be incorporated which not only render support for planning and plan execution as empasized in the past. Instead, main emphasis will be placed on autonomous machine situation assessment in parallel to the crew's situation assessment activity which leads to better machine understanding of what the real needs of the crew are and consequently to more efficient support for the sake of flight safety as well as mission effectiveness. There are already examples of successful developments, which have proven that the way of design guideline implementation as described in this paper systematically led to the desired system performance. One example is the Cockpit ASsistant System CASSY. The successful flight tests of CASSY in real IFR flights have demonstrated that human-centered automation by means of intelligent onboard systems can be integrated into the cockpit of modern aircraft. Speech recognition proved to be a powerful device as pilot interface, but other devices should also be considered. Optimal performance can be achieved with a digital data link between ground and airborne segments. The amount of detected and avoided pilot errors, the availability of features like pilot intent recognition as well as the power demonstrated in complex planning indicate the performance level of CASSY. This kind of system can be considered as the coming solution to overcome existing criticism with respect to current flight management systems.

REFERENCES

Adam, V., Klostermann, E. and Schubert, M.: DLR's ATM Demonstration Programme. AGARD Meeting on 'Machine Intelligence in Air Traffic Management', 1993.

Amalberti, R. and Deblon, F.: "Cognitive Modelling of Fighter Aircraft Process Control: Step towards an Intelligent Onboard Assistance System", Intern. Journal on Man Machine Studies, 36. Academic Press, 1992.

Billings, C.: "Human-Centered Aircraft Automation: A Concept and Guidelines", NASA Tech Memorandum 103885, Moffet Field, 1991.

Bittermann, V., Perennec, M., Mielnik, J.-C., Sainthuile, G. and Jud, J.-M: "FINDER: Flight-Plan Inetractive Negotiation and Decision-Aiding System for Enroute Rerouting", 12th Conf. Intelligence Artificielle, Systemes Experts, Langage Naturel, Avignon, June, 1992.

Boy, G.: "Intelligent Assistant Systems", Academic Press, London, 1991.

Chambers, A.B. and Nagel, D.C.: "Pilots of the Future: Human or Computer?", Comm. of the ACM, Vol.28, No.11, Nov. 1985.

Dudek, H.-L.: "Wissensbasierte Pilotenunterstützung im Ein-Mann Cockpit bei Instrumentenflug", Dissertation, UniBw, Munich, 1990.

Dyck, J.L., Abbott, D.W. and Wise, J.A.: "HCI in Multi-Crew Aircraft", Proc. of the HCI, Orlando, 19 A, 08/93, 1993.

Endsley, M.R. and Bolstad, C.A.: "Human Capabilities and Limitations in Situation Awareness", Proc. NATO AGARD FMP/GCP Symposium, Edinburgh, UK, 1992.

Gerlach, M. and Onken, R.: "A Dialogue Manager as Interface Between Aircraft Pilots and a Pilot Assistant System", Proc. of the HCI, Orlando, 19th A, 08/93, 1993.

Harris, D.F. and Morrissette, J.A.: "Single Pilot IFR Accident Data Analysis", NASA CR 3650, 1982.

Helander, M., Moody, T.S. and Joost, M.G., "Systems Design for Automated Speech Recognition", in Helander (ed.), "Handbook of Human-Computer Interaction", Elsevier Science P., 1988.

Heldt, P.-H.: "Flying the Glass-Cockpit", DGLR-Fachausschuß Anthropotechnik, Berlin, 1993.

Hollnagel, E.,Warren, C. and Cacciabue, P.C.: "Automation in Aerospace and Aviation Applications: Where are the Limits?", WS. Rep. 4th HMI-AI-AS, Toulouse, 09/93, 1993.

Huber, P., Jensen, K. and Shapiro, R.M.: "Hierarchies is Coloured Petri Nets", in: Rozenberg, G. (ed.): Advances in Petri nets 1990, Lecture Notes in Computer Science, vol. 483., Springer, Berlin, Heidelberg, New York, 1990, 313-341.

Huck, V.: "Petri-Netze im Flughafenverkehr - Möglichkeiten, Grenzen, Perspektiven", DLR-Mitteilung 91-17, Braunschweig, 1991.

Kreher, H.-J., Stöhr, P. and Schroer, W.: "Simulationsmodelle zur Analyse des Bodenrollverkehrs auf Flugplätzen", in: Entwurf komplexer Automatisierungssysteme, TU Braunschweig, 1993.

Lloret, J.C., Roux, J.L., Algayres, B. and Chamontin, M.: "Modelling and Evaluation of a Satellite System Using EVAL, a Petri Net Based Industrial Tool", in: Jensen, K. (ed.), Application and Theory of Petri Nets 1992, Lecture Notes in Computer Science, vol. 616, Springer, Berlin, Heidelberg, New York, 1992, pp. 379-383.

Meta Software Corporation: Design/OA, Version 3.0, Cambridge, 1991.

Nilsson, N.J.: "Principles of Artificial Intelligence". Springer Verlag, 1992.

Norman, D.A.: "The 'problem' with automation: inappropriate feedback and interaction, not 'over-automation'", in: Human Factors in Hazardous Situations (D.E. Broadbent, A. Badderly and J. T. Reason (Ed.)) pp. 585-593. Clarendon Press, Oxford, 1990

Onken, R.: "Automatic Pilot Aid for Situation Awareness and Flight Planning", HMI-AIAS, Toulouse-Blagnac, Sept. 1988.

Onken, R.: "New Developments in Aerospace Guidance and Control: Knowledge-Based Pilot Assistance", 12th IFAC Symposium on Automatic Control in Aerospace, Ottobrunn, 1992.

Onken, R.: "Funktionsverteilung Pilot-Maschine: Umsetzung von Grundforderungen im Cockpitassistentsystem CASSY", DGLR-Tagung des Fachausschuß Anthropotechnik, 1993.

Onken, R. and Prévôt, T.: "CASSY-Cockpit Assistant System for IFR Operation", ICAS 94-8.7.2 Proceedings Vol.2, Anaheim, CA, 1994.

Onken, R.: "Human-centered Cockpit Design through the knowledge-based Cockpit Assistant System CASSY", NATO DRG Panel 8 Workshop on improving function allocation for integrated system design, Soesterberg, 1994.

Prévôt, T., Onken, R. and Dudek, H.-L.: "Knowledge Based Planning for Controlled Airspace Flight Operation as Part of a Cockpit Assistant", AGARD GCP Symposium, Amsterdam, 1991.

Prévôt, T. and Onken, R.: "On-Board Interactive Flight Planning and Decision Making With the Cockpit Assistant System CASSY", 4th HMI-AI-AS, Toulouse, 1993.

Proceedings of the 3rd International Workshop on Human-Computer Teamwork, "The Human-Electronic Crew: Can we trust the team". Cambridge, UK,1994.

Rasmussen, J.: "Skills, Rules, and Knowledge: Signals, Signs, and Symbols, and other Distinctions in Human Performance Models", IEEE SMC-13 (3), pp 257-267, 1993.

Rasmussen, J.: "Information Processing and Human Machine Interaction An Approach to Cognitive Engineering", Elsevier, Amsterdam, 1986.

Reisig, W.: "Petri Nets and Algebraic Specifications", in: Theoretical Computer Science 80, 1991, pp. 1-34.

Reising, J.M., Emerson, T.J. and Munns, R.C.: "Automation in Military Aircraft," Proc. of the HCI, Orlando, 19A, 08/93, 1993.

Ruckdeschel, W. and Onken, R.: "Modelling of Pilot Behaviour Using Petri-Netze", 15th Int. Conf. on Application and Theory of Petri Nets, Zaragoza, 1994.

Rummelhart, D.E and McClelland, J.L.: "Parallel Distributed Processing. Explorations in the Microstructure of Cognition", MIT Press, Cambridge, MA, 1987.

Starke, P.: "Analyse von Petri-Netz-Modellen", B.G. Teubner, Stuttgart, 1990.

Starke, P.: "INA Integrated Net Analyzer", Version 1.3, Berlin, 1993.

Strohal, M. and Onken, R.: "Cockpit Assistance", Human Factors Workshop, IABG,1994.

Taylor, M.M. and Waugh, D.A.: "Principles for integrating voice I/O in a complex interface", AGARD Avionics Panel Symposium, Madrid, Spain, May 16-22, 1992.

Vogler, W.: "Modular Construction and Partial Order Semantics of Petri Nets, Lecture Notes in Computer Science", vol. 625, Springer, Berlin, Heidelberg, New York, 1992.

Wiener, E.L. and Curry, R.: "Flight Deck Automation: Promises and Problems", Ergonomics 23, 1980, pp 995-1011.

Wiener, E.L.: "Human Factors of Advanced Technology ("Glass Cockpit") Transport Aircraft", NASA Report 177528, Moffett Field, CA, 1989.

Wiener, E.L.: "Error Exposure and Control in High Technology Aircraft", in: Artificial Intelligence in Cockpit Design, Orly, 1989.

Wilensky, R.; "Planning and Understanding", Addison Wesley Reading, Massachussetts, 1983.

Wise, J.A., Abbott, D.W., Tilden, D., Dyck, J.L., Guide, P.C. and Ryan, L.: "Automation in Corporate Aviation: Human Factors Issues", Interim Technical Report CAAR-15405-93-1, FAA, 1993.

Wittig, T. and Onken, R.: "Pilot Intent and Error Recognition as a Part of a Knowledge Based Cockpit Assistant", AGARD Joint FMP/GCP Symposium, Edingburgh, 10/92, 1992.

Wittig,T., Onken, R. and Zeller, H.: "Entwicklungsperspektiven intelligenter Flugführungssysteme", DGLR-Jahrestagung, Erlangen, 1994.

Wittig, T.: "Maschinelle Erkennung von Plotenabsichten und Pilotenfehlern über heuristische Klassifikation", Dissertation an der Universität der Bundeswehr München, 1994.

Zadeh, L.A.: "Fuzzy Sets", Information and Control, Vol. 8, 1965.

Development environment for Knowledge-Based Systems Some examples of Application : The "Copilote Electronique" project

G.Champigneux

DASSAULT AVIATION
78 Quai Marcel Dassault, 92214 Saint Cloud, France

1 SUMMARY

This paper aims at describing first lessons learnt in terms of engineering guidelines and development methodology within the french project called "Copilote Electronique" of an Electronic Crew Member System.

This project is lead by Dassault Aviation with the support of French official services (DRET, STTE) and involves several industrial and scientific partners (SAGEM, Dassault Electronique, Matra Defense, Sextant Avionique, IMASSA).

The French "Copilote Electronique" project has started in 1986 through various preliminary studies and since 1994 it took a larger scale under the form of an exploratory development.
Before the start of this development, advantages and drawbacks of existing software engineering or knowledge acquisition methodologies have been compared. Emphasis has been put on ergonomics design rules and on a project life cycle adaptation aiming at insuring better responses to pilots demands and fears...
Building on the first year experience of the exploratory development phase of the Copilote Electronique project, we express confidence for successful operational evaluations.

2 INTRODUCTION

In spite of great benefits expected, Knowledge Based System (KBS) approaches are not so easy to apply in the pilot assistance field.

Since 1986 the french official services have supported several studies in the field of Pilot's assistance. The technical push of Artificial Intelligence and Knowledge Based System combined with the results of cognitive analysis of pilots activities resulted in the recognized need in the Guidance and Control community for more support to the Pilot in complex missions [*Banks 91*].

In France, it gave birth to the french concept called "Copilote Electronique" [*Champigneux 89*]. This assisting system relies on several in flight mission planning activities organised in a cooperative architecture.

We present the first lessons learned through the preliminary stages of the "Copilote Electronique" operation as well as the first active year of the exploratory development phase.

The paper is divided into five main sections:

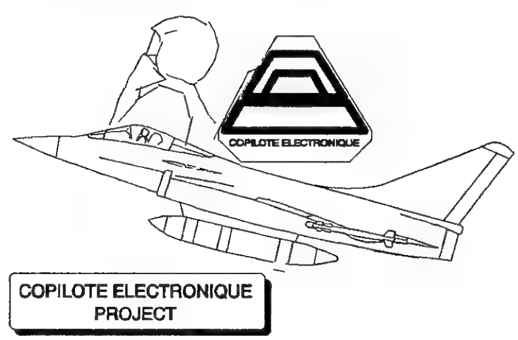
- In section 3 we describe general goals and characteristics of the project.

- In section 4 we first recall design difficulties encountered in such projects : symbolic programming is not magic, knowledge acquisition is not so easy, process automation has to keep user or pilot in the loop, internal and external co-ordination problems arise and specific ergonomics rules are to be applied.

- In section 5 we propose a discussion on rapid prototyping life-cycle model; its advantages, known drawbacks and possible solution from classical software engineering field. The knowledge acquisition and software engineering life cycle chosen for "Copilote Electronique" is illustrated.

- In section 6 we summarise first lessons learnt during the preparation phase of the project and the first year of the new development phase.

- In the last section we open to the future developments of the Copilote Electronique project.



Paper presented at the Mission Systems Panel of AGARD and the Consultant and Exchange Program of AGARD held in Madrid, Spain, on 6-7 November 1995; Châtillon, France, 9-10 November 1995; and NASA AMES Research Centre, California, USA, 16-17 November 1995 and published in LS-200.

3 THE "COPILOTE ELECTRONIQUE" PROJECT

System and software design methods that have been used for current generation fighters are facing more and more difficult challenges and may encounter their own limits within ten or twenty years from now.

Various embedded functions, such as navigation, piloting, aircraft status management, weapons system management, and in some extension sensors management have been successfully automated by classical software engineering methods, but the addition of such separate and independent automated functions become more and more difficult to control in real time situations by human pilots.

Nevertheless, as critical decisions are to be taken on uncertain or tactical aspects of mission, aircraft designers often rely on pilots judgement. This tendency is even currently required by Air Forces.

As automated functions are intended to increase in number and **complexity**, in the foreseen tactical context characterised by a great number of various possible threats, with electronic war systems and new sophisticated weapons, operational experts think that future pilots will have some difficulties with this combinatory explosion of information sources unless being assisted in their reasoning tasks.

An expert assistance system, will have to absorb high rates of raw information, select and highlight the more crucial ones, before initiating dialogue, in a manner adapted to current situation and mental load of the pilot. Such a system should only present **pertinent information** and offer a restricted actions choice to the pilot, on which, after selection by the pilot, it will have to examine all consequences before execution.

The "Copilote Electronique", initialised in 1986 by the French DGA ("Délegation Générale de l'Armement"), aims at introducing, within a 2010 horizon, expert or knowledge based systems in combat aircrafts. Far from replacing human pilots in the cockpits, such a sophisticated electronic crew member should be considered as a very **high level dialogue function** between man and machine.

In fact, this project took a new acceleration in 1994 spring, when the Technical Service for Aeronautical Telecommunication and Equipments of French DGA (STTE) decided the funding of an exploratory development for RAFALE standard SU2, which is the Rafale standard that will enter French Air Forces in 2004. SU2 standard will benefits of all radar modes, counter measures and a front infrared detection and tracking system; besides pilot will use helmet mounted displays.

The goal of this exploratory development phase, launched for a three years duration, is a ground simulation, without hard real time constraints, to demonstrate the "Copilote Electronique" in situation of strike and escort missions, with low altitude penetration constraints.

4 DIFFICULTIES FOR PILOT ASSISTANCE SYSTEM DESIGN

The complexity of problems in the aeronautic field led some designers to propose Knowledge Based System (KBS) methods as an easier and more comfortable programming solution than "classical laborious and error prone" software development.

With new symbolic software environment, developers quickly produced promising prototypes in view of production system delivery. But as no specific software engineering methodologies were generally applied, it became obvious that desirable high quality and maintainable systems were not reachable. Since then a relative disillusion is felt in the aeronautic community.

To understand KBS interest and complexity of development, one has to take into account the importance of human expertise in the design process. Human experts need to be considered as full members of the project team involved at each stage of the development : early description of the problem domain, requirements definition, design, description of performed tasks, ideas of new man-machine dialogue, validation, end-use,... This central role influences the developing environment and suggests the modification of the infrastructure. KBS pretends to level up the knowledge representations so that the human specialists can understand what is in the machine. It requires new concepts like objects, plans, heuristics, agents... The expert may even want to flip from one representation to another. So that, knowledge engineers have to walk their way through a very large set of representation schemes.

Pilot Assistance systems must present specific characteristics : they must be real-time systems (involving most of the time some temporal reasoning), embedded on-board aircrafts (satisfying CPU, memory, ergonomical... restrictions), most frequently multi-expert, deeply integrated into their environment and keeping the end-user in-the-loop.

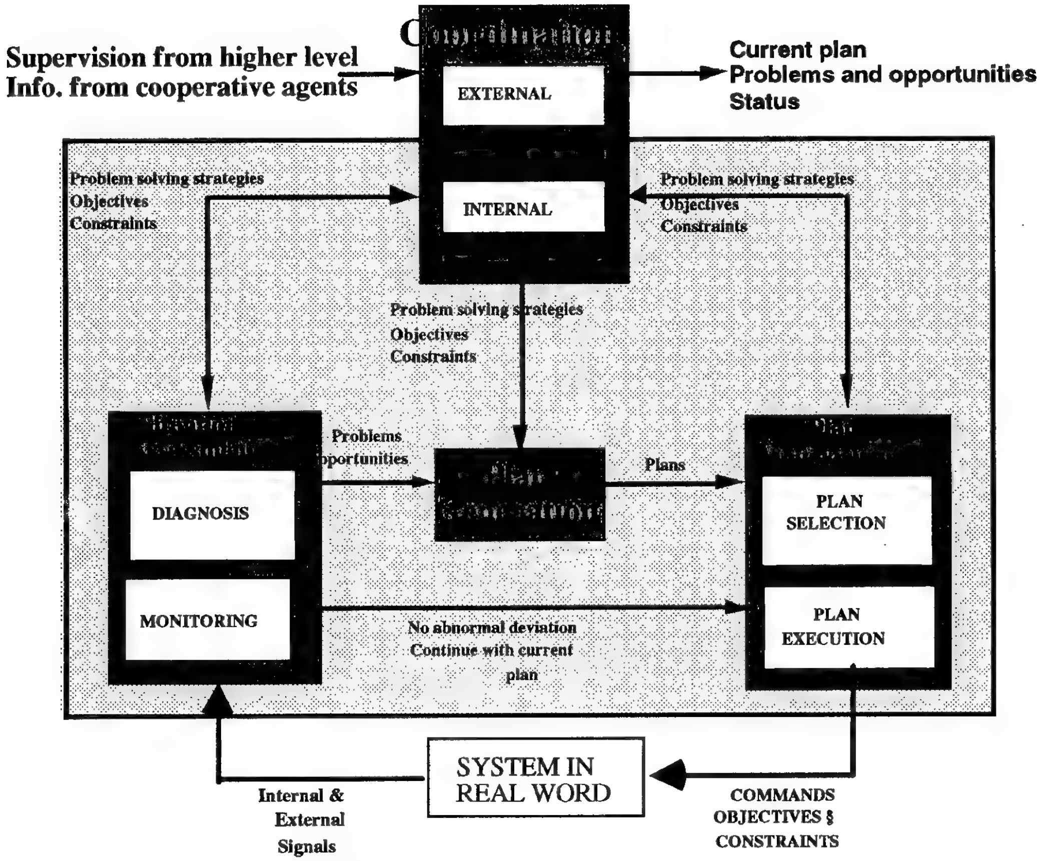


Figure 1 : Guidance and control system functional decomposition

In the report of the AGARD working group 11 [*AGARD 1993*], a functional analysis of Knowledge Based Systems in Guidance and Control (G+C) field is stated (figure 1). It is quite adapted to pilot's assistance field. In particular, the classification in three main engineering domain is pertinent in pilot's assistance : situation assessment (monitoring, diagnosis), plan management (plan selection and execution), co-ordination (external and internal).

A more detailed discussion on specific difficulties for intelligent assistance design in G+C field can be found in [*Sallé 1993*].

Non respect of ergonomics rules is the most current explanation for KBS applications failures, in the field of ind·strial processing assistance.

In the "Copilote Electronique" team this difficulty is taken in charge by IMASSA (Centre for Medical Studies and Research in Aerospace) to define "user oriented rules" that has to be used from the design phase [*Amalberti et al 1990*].

Those rules can be summarised as follows :

• pilot anticipates and needs anticipation assistance on contrary of "classical engineer designed" assistance which are often too reactive,
• pilot's decisions reflect often compromises between mental load and ideal response to the situation, so pure optimality is not to be researched if pilot has no sufficient time to understand,
• following their own personal skills, different pilots may organise work differently, assistance must be adapted to these skills,
• assistance must be homogeneous, and it will be preferable to rely on specialised expert for each operational domain (e.g. Strike or Air Defense expertise) so resulting assistance will produce constant understanding interpretation model that will avoid surprises for pilot,
• assistance must know and respect its own limits,
• system design may use "what if" approach to be less reactive,
• dialogue must be adapted to context, pilot intents and pilot load,
• dialogue must be space oriented and interactive, better use vocal media than written, but avoid saturation,
• respect logic of pilot understanding, that means rely on the understanding model designed with expert pilots.

4 SOFTWARE LIFE CYCLE & KBS METHODS FOR THE COPILOTE ELECTRONIQUE

The use of rapid prototyping to build a complete system, using one of the many software packages available on the market, has been a frequent technique in the early studies of the "Copilote Electronique" project. This method also known as "evolutionary prototyping" or "iterating prototyping" was deduced from experimental approaches to KBS development.

It consists in iterating the cycle :

•knowledge acquisition,
•implementation : knowledge modelling & coding,
•validation,
•test with the expert(s), until there is no more knowledge to capture.

Detailed discussion on obvious advantages but also on subtle drawbacks of this life-cycle methodology can be found in [*Sallé 1993*]. This rapid prototyping methodology is not relevant for complex and embedded systems for which a good architectural design is needed.

As a consequence of risks encountered in final system integration phase, when rapid prototyping method is followed, alternate methods were looked for the "Copilote Electronique" project.

Some theoretical studies tend to consider the KBS as an ordinary software production problem with its overall analysis prior to any implementation. KADS is the leader of this new way of design [*Hickman et al 1989*], MOISE [*Ermine 1992*] is another example. Based on a model driven-approach, these methodologies have much in common with conventional software development methodologies : they prescribe phases, stages and activities, models, documents and deliverables (figure 2).

To maintain the benefits of dialogue with experts, a KBS approach iterating design illustrated by a spiral model was introduced [*Hickman et al 1989*]. This model may be considered as an overall life-cycle model and may be inserted at the top of the inverse V-model depicted (figure 2). We use this spiral model to show all efforts made in the early phases of "Copilote Electronique", before the Exploratory Development phase (figure 3).

Current knowledge engineering practices heavily depends on interview techniques and the collection and analysis of notes. The process, although valuable, is slow and frequently paces the development activity.

There is general agreement that considerable gains in speed and efficiency can be achieved by improving both tools and methods. But additional efforts, for software designers, in knowledge acquisition and knowledge elicitation from expert pilots are not to be minimised, we think, as it is the **main benefit of KBS oriented pilot assistance design.** This can be illustrated by the climbing branch of figure 2 inverse V-model (following some ideas from [*Dieng 1990*]), this climbing branch being more important than in classical V-cycle, where existence of a clear specification is always the starting point of process and future disappointments.

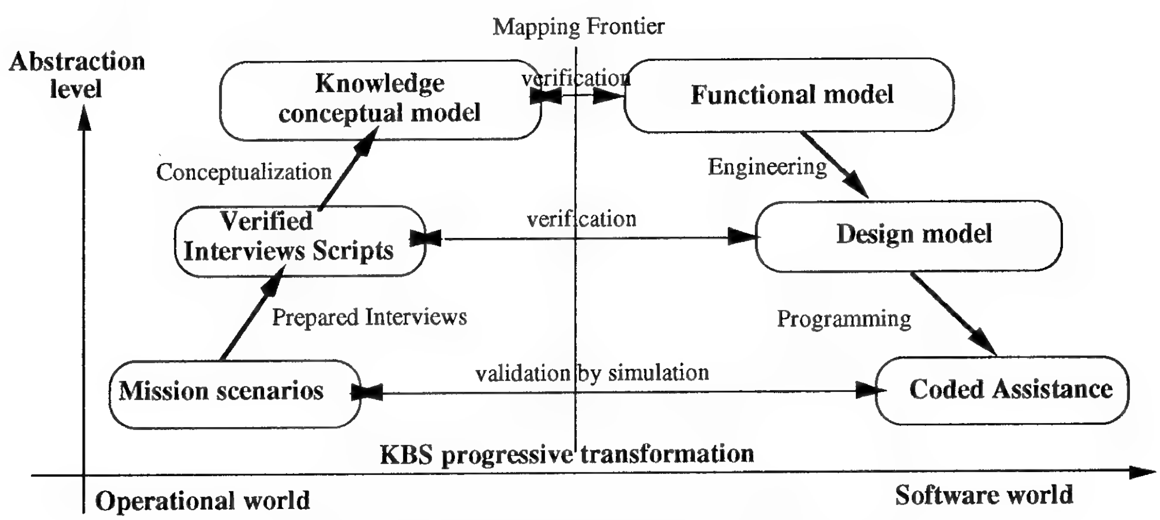


Figure 2 : COPILOTE ELECTRONIQUE V-cycle

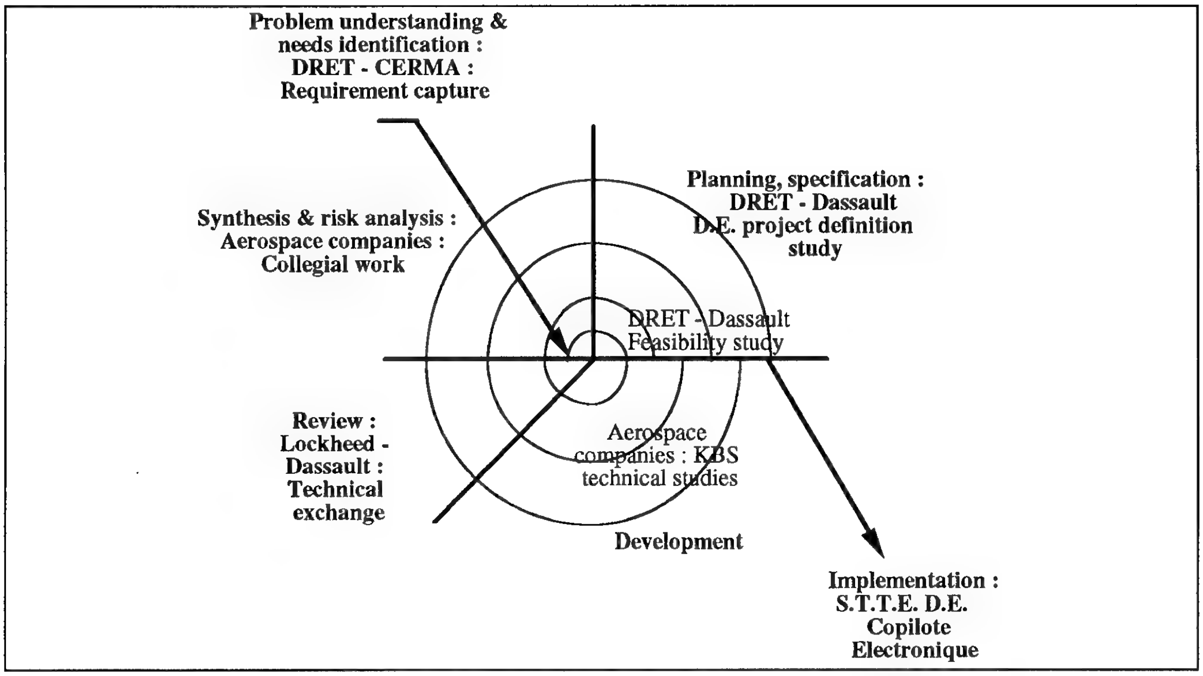


Figure 3 : Spiral Model applied to Copilote Electronique

5. LESSONS LEARNT IN "COPILOTE ELECTRONIQUE" PROJECT

At the end of the phase of problem understanding and needs definition of the "Copilote Electronique" program (figure 3), three main risks were identified :

• Is it possible to capture enough expertise to create a real assistance for pilot reasoning ?
• Is the KBS technology mature enough for a real development ?
• Is the French Aerospace community able to integrate such a new concept in a current avionics system design ?

A survey of existing efforts (national as well as international) was initiated to give answers to these questions. An extensive work sponsored by DRET (French Defence Advanced Research Agency) and realised by IMASSA (Centre for Medical Studies and Research in Aerospace), with Mirage F1 Recce pilots, gave a lot of clues resulting in the identification of various pilots behaviour correlated with pilots profile. It brought to the front scene the pilot's "meta-knowledge" (specialised technical education) which influences mission planning as well as reflex reactions.

A joint work with Dassault allowed to map pilots intuitive aspiration for new assistance with the reality of Mirage 2000 and Rafale advanced design.This analysis conducted by Dassault Aviation with the support of the french official services DRET prooved the feasibility of the "Copilote Electronique" concept and established the proper methodological lines for the exploratory development.

In order to achieve the main objective of demonstrating the concept of a crew assistant for future combat aircraft it is necessary to organize expert entities that will perform the required functionalities of in flight planning.

The Copilote Electronique project finalized such an architecture.

The architecture of the Copilote Electronique is turned to a focused application i.e. single seater combat aircraft in an air to air escort role and/or an air to ground bomber role.

The top level organization of the expert entities in the Copilote Electronique is in accordance with the Functional Decomposition of Generic decision system in Guidance and Control as proposed by AGARD Working Group 11.

The two main activities of Situation assessment and planning are represented by two layers of reasoning called "reflexion" and "decision".

The coordination activity is taken in charge by a specific entity called "Information Manager".

We can have a Copilote Electronique Organic View of the decision making systems by considering planning in connection with perception assessment communication and execution (figure 4).

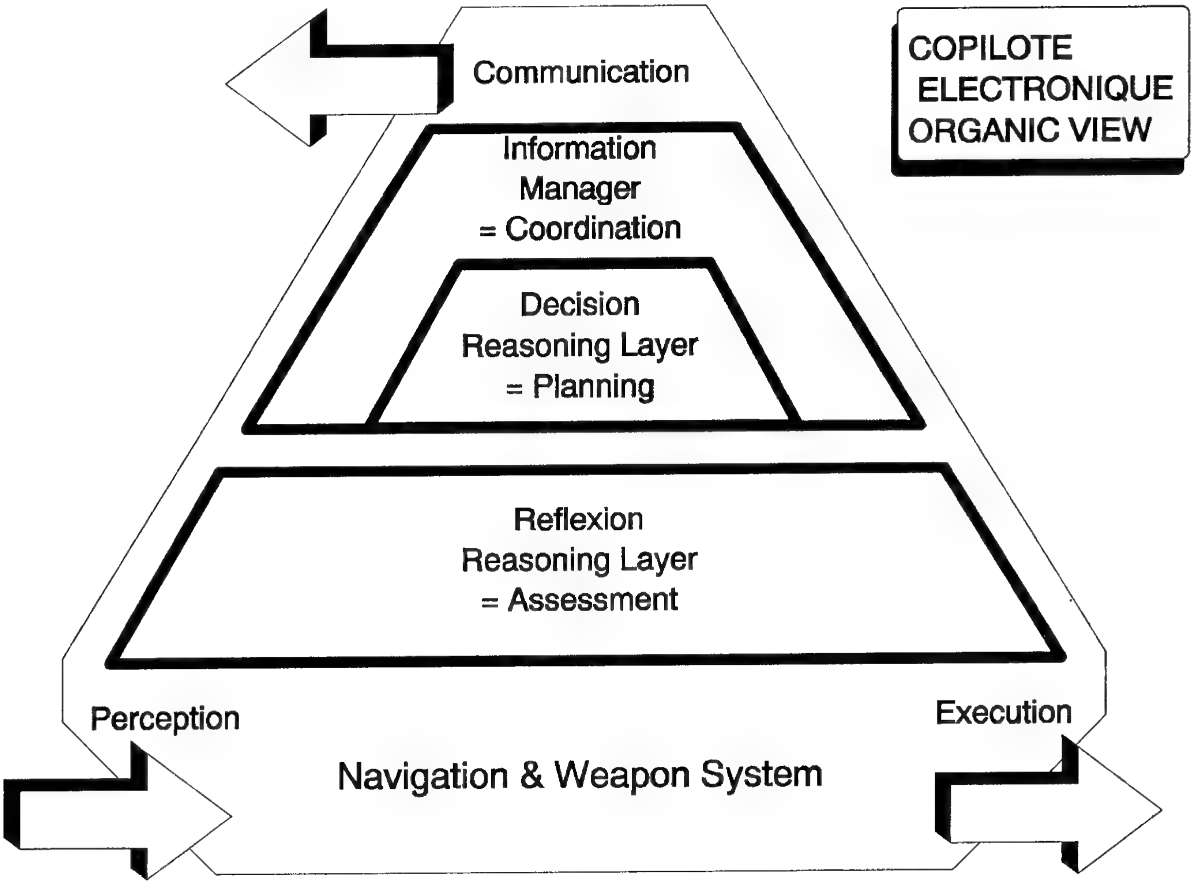


Figure 4 : Copilote Electronique Organic View

The planning reasonning layer will take entries from the assessment level. Those entries are problems , alarms, situation descriptions and hypotheses... This layer will perform plan generation and plan selection .

These generic architectural views can be instanciated on the specific domains of planning required by the decision process in combat missions. This limits the scope of the functionnalities and allows at this level of detail the selection of proper mecanisms.

The external world perception, the communication with other agents and the plan execution are not part of the Copilote Electronique responsibility but we can assume that these activities are present in the current Navigation and Weapon system (SNA) in which the Copilote Electronique is integrated .

Planning tasks are certainly of the type requiring a good cooperation between man and machine in order to keep the "Pilot in the Loop". We structured those tasks like system reconfiguration, ressources scheduling, navigation, fuel monitoring threat analysis, threat avoidance, threat engagement, Command Control and Communication, sensor control and weapon management into three main areas :

- System management
- Tactics managment
- Mission management

In the System area, planning is more often an optimization of fine grain plan in front of the flight parameters evolution and generalised state of the navigation and weapon systems (including faulty states).

In the Tactics area, planning is reactive. Threats are poping up as unexpected event and disrupt from the planified behavior established on ground during the preparation phase.

In the Mission area, the result of the mission preparation remains the guide for all in flight planning. The task here consists of local adaptations of the nominal plan, plan refinement in a precise context, choice of alternative plans...

In consequence of this planning analysis, assistance in the Copilote Electronique project is split between what is called Expert assistance domains.

These Expert domains are:

-system planning,

-tactical planning
similarly split into:
-air threat reactive planning,
-ground threat reactive planning and
-perceptive planning.

-mission planning.

This decomposition is illustrated in the "pyramid" view of the Copilote Electronique Architecture (figure 5).

Information required by in flight mission planning consist in :
-vehicle performance information such as aircraft navigational accuracy, altitude measurement accuracy, endurance, equipment status including health evaluation,
-threats identifications locations and forecasted evolutions as well as friendly forces situation,
-mission information such as the target areas description for reconnaissance and attack, geographical data over the mission paths and weather data on mission legs.

Each expert domain can in turn be refined into precise planning task (or task necessary to support the planning activity).

An initial decomposition is summarised in the following list :

- System planning

  . Planning the avionic sy^tems reconfiguration.
  . Scheduling of action & ressources according to the plan.
  . Assessing efficiency and feasibility of the plan.

- System Evaluation

(in support of system planning)

  . Monitoring discrete events.
  . Monitoring continuous signals.
  . Assessing real avionic systems states and dependability.

- Tactical planning

(similar decomposition of air threat reactive planning and ground threat reactive planning, at this point a finalized decomposition of perceptive planning has not been achieved)

  . Planning tactics according to the threats.
  . Scheduling of action & ressources according to the tactics.
  . Handling conflicts among proposed tactics.

- Tactical assessment

(in support of tactical planning)

  . Analysis of friendly & foe forces.
  . Elaboration of forecasted evolutions.
  . Assessment of risk/efficiency according to present plan.

- Mission Planning

  . Selecting re-routing options accoring to the updated mission context.
  . Planning new routes.
  . Monitoring possible routes with quality estimates.
  . Selecting current route.

- Mission condition Assessment.

  . Mapping of pre-mission meteorological brieffing onto possible routes.
  . Mapping of pre-mission geographical data onto possible routes.

The in flight planning mecanisms will be dynamically controlled based on a human-centered planning paradigm. This is realized in the Copilote Electronique through the following planning control modules :

- Pilot-Planning Assement.

  . Monitoring Pilot actions.
  . Assessing Pilot Intents.
  . Assessing Pilot Workload.

- Assistant planning assessment.

  . Monitoring information quality according to the plan.
  . Monitoring current plannification according to the mission requirements.
  . Monitoring the plan execution.

- Pilot<->Assistant coordination.

  . Allocating tasks to man or expert agents.
  . Monitoring MMI ressources.
  . Solving conflicts among expert agents.

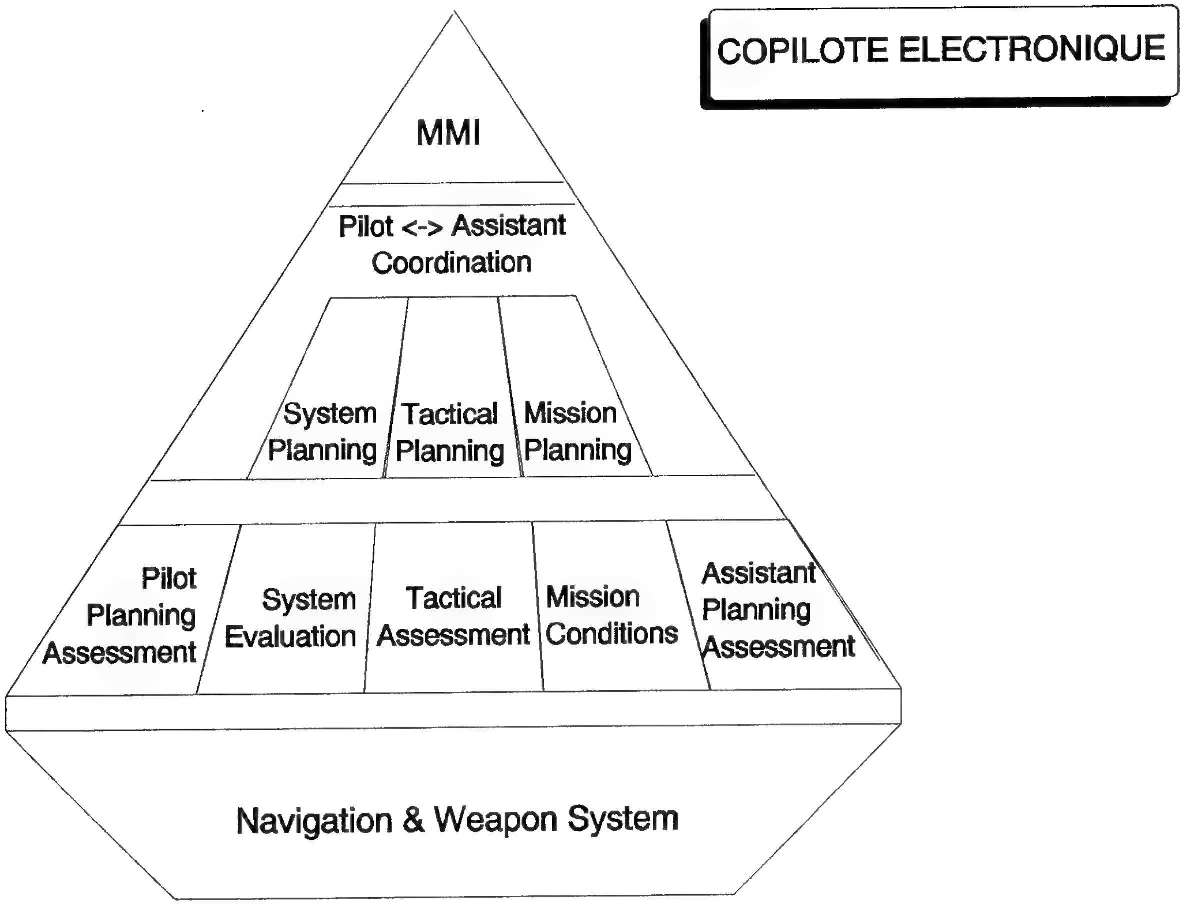


Figure 5 : Copilote Electronique Architecture

The proposed architecture at this stage is resolutely a cooperative set of expert domains.

It directly map the french industrial competences, and permit to select a viable consortium for the Exploratory Development phase (figure 6).

•System Status Assessment and Management	->	SAGEM
•Tactical Situation Assessment and Management (Ground threat and defensive Counter Measures)	->	DASSAULT ELECTRONIQUE
•Tactical Situation Assessment and Management (Air threat and offensive Weapons)	->	MATRA DEFENSE
•Mission Conditions Assessment and Mission Management	->	SEXTANT AVIONIQUE
•Pilot Assessment, action plans assessment, relevant information management and man machine interface	->	DASSAULT AVIATION
• Ergonomics rules and knowledge acquisition methods and verification tasks	->	IMASSA

Figure 6 : Exploratory Development Consortium

Several actions were initiated to reduce foreseen difficulties. An international co-operation was carried by Dassault Aviation with Lockheed in order to confront the French approach with the US team experience and avoid traps experimented in previous experiments [*Smith et al 1988*], [*Rouse 1991*]. Our experience in the development of the Copilote Electronique as well as the technical exchanges with Lockheed Pilot's Associate team, led to the fact that conventional knowledge engineering techniques using questionnaires and interviews are not sufficient **to provide implementable and secured knowledge for Pilot aids.**

The investigated knowledge engineering techniques, in order to tackle the problem of such a complex AI system development, should be used for expertise initial design, then supplemented by extensive knowledge evaluation and correction in simulator. With IMASSA, a specific method for eliciting and formalising pilot's expert knowledge was studied and is used. It is supported by a formalisation tool called X-PERT. A considerable amount of expertise has been acquired through interviews with four pilots. It concern both air to ground penetration missions and air to air escort missions. All the interviews are recorded and fully transcripted in text form. All the set of transcripts is used by the knowledge engineers of the industrial and scientific partners of the Copilote Electronique exploratory development.

Our technical specification is driven toward a flexible heterogenous implementation paradigm. Our choice is to organize the modules in a multi-agent system using Distributed Artificial Intelligence techniques [Erceau 1991].

Another very important technical issue is the definition of a common **"plans and goals" exchange language** between all specific assistance modules, and great efforts are to be made to maintain this common message glossary. With-in the Exploratory development phase Dassault Aviation proposed an exchange language called LDI which provides a CORBA like facility for object communication.

Finally, a unifying technical principle was adopted to facilitate the architecture design via the **intent planning paradigm**. This principle is essential to fulfill general ergonomics constraints : assistance must not participate to the signalled existing overloading factors. Intent recognition is a challenging but promising direction and can be made easier by extended preparation mission plans and procedures (for each pilot activity) that will be perhaps the new "automated and personalized" check lists version of the future.

7. CONCLUSION

The technology is available today to provide viable knowledge system solutions to well-chosen and well-defined problems. It can be expected to see more and more successful projects on such on-board applications, as both the research, the technology and engineering skills of application developers improve.

But this process may be slower than was though. Main reason is that knowledge acquisition tasks and user oriented ergonomics rules compliance must be integrated in the overall engineering cycle.

The french Copilote Electronique project has been carefully planified considering those methodological difficulties.

After a long design phase the Copilote Electronique is now in a software development phase. The planning domains are the main drivers of this development. They are studied by various partners in a federative approach. Each partner brings to the project a specific background, with a high value knowledge of his planning field and mastering of appropriate planning mecanisms. This results in a very rich but heterogeneous multi-expert, multi-industrial planning system.

Our aim is to, not only reach a successful behavior in each planning field, but also to achieve a coherent assistant for in flight planning. Planning proposals will be demonstrated on a realistic full mission simulator. Special care is taken to analyse interdependancies between the various plans and to respect the rules of a good man machine relationship. Expert pilots will give feedback on the quality and acceptability of the resulting planning assistant. According to their remarks the architecture, mecanisms and knowledge of the Copilote Electronique planners will be tuned.

Real time performances of the resulting planning system will be optimised next with the help of current technological progress (specially modular avionics and new software environment). Our belief is that the key of a successful in flight planning is more in the pilots cognitive abilities than in hardware/software evolution.

The first steps of the Exploratory Development phase confirms that the distributed architecture and the Human driven design approach are good drivers for success. The consortium is now entering the multi-expert assistant prototyping with confidence on the resulting operational benefits.

ACKNOWLEDGEMENTS

This paper has been prepared with the help of Working group 11 members and special acknoledgements should be send to Stéphane Sallé.

We would like to thank also the complete Copilote Electronique team, and in particular, Thierry Joubert from Matra Défense, Stéphane Sallé from SAGEM, Pierre Sassus from Sextant Avionique, Hervé Le Doeuff from Dassault Electronique and Jean Yves Grau from IMASSA.

REFERENCES

[*Banks 91*] : **S.B.Banks, C.S.Lizza,**
"Pilot's Associate",
IEEE Expert V.18 pp18-29 1991.

[*Champigneux 89*] : **G.Champigneux, Y.Gaudry, A.Gilles, P.Nory, M.d'Orso,**
"Vers une Intelligence Artificielle embarquée pour l'assistance au pilote: le Copilote Electronique",
Conférence IA et Défense AVIGNON 1989.

[*AGARD 1993*] : **AGARD Working group 11**
Knowledge Based System Approaches for Guidance and Control",
AGARD Working group 11 report published in spring 1993.

[*Sallé 1993*] : **Sallé S.E., Champigneux G., Reichert O., Gaudry Y.**
"Development Methodology for Knowledge-Based Systems used in Guidance and Control : application to the COPILOTE ELECTRONIQUE project",
13_i_th International Conference on Expert Systems and AI, AVIGNON 93, AI and Defence conf, EC2, 1993.

[*Amalberti et al 1990*]: **Amalberti René & Deblon François**
"Cognitive modelling of fighter aircraft process control : a step towards an intelligent onboard assistance system",
Int. J. of Man-machine studies.

[*Hickman et al 1989*] : **Hickman et al.**
"Analysis for knowledge based systems. A practical guide to the KADS methodology",
Ellis Horwood Ed., 1989.

[*Ermine 1991*] : **Ermine Jean-Louis**
"Aspects méthodologiques et industriels des systèmes à base de connaissances",
Journées état de l'art du GRECO-CNRS, 1992.

[*Dieng 1990*] : **Dieng R.**
"Méthodes et outils d'acquisition des connaissances",
ERGO-IA congress ,Biarritz (Fr), 1990.

[*Smith et al. 1988*] : **D.D.Smith, M.Broadwell,**
"The Pilot's Associate: an overview",
International AI Conference AVIGNON 1988.

[*Rouse 1991*] : **Rouse W.**
"Design for success : A human centered approach to designing successful products and systems",
Wiley series in systems engineering, 1991

[*Erceau 1991*] : **J.Erceau, J Ferber,**
"L'intelligence Artificielle Distribuée",
La recherche 233 Vol 22 pp750-758 1991.

KNOWLEDGE-BASED SYSTEMS APPLICATIONS FOR AUTOMATING SPACECRAFT OPERATIONS: USERS FEEDBACK & LESSONS LEARNED.

Jean-Michel DARROY, Arnaud DE SAINT VINCENT, François LECOUAT
MATRA MARCONI SPACE
Z.I. DU PALAYS
31 RUE DES COSMONAUTES
31077 TOULOUSE CEDEX, FRANCE

1 - SPACECRAFT MISSION OPERATIONS

This section outlines a baseline structure for knowledge-based guidance and control functions in the context of space missions (ref 3). Guidance and control functions will be referred to here as "Spacecraft Mission Operations", i.e., all the functions required to implement space missions. The development of a general baseline structure is difficult since these functions may vary considerably from manned space mission to unmanned mission, or from an earth observation satellite mission to a communications satellite mission.

This section focuses on unmanned missions since they correspond to a majority of the actual space missions, and it presents a generic structure that is applicable to any kind of unmanned mission. Adaptation of this structure to manned space missions (e.g. Space Shuttles, Space Stations) is also discussed.

1.1 - Functional Structure of Spacecraft Mission Operations

A Spacecraft is generally composed of two main parts :

The payload

Composed of any on-board equipement directly associated with the mission itself including : observation instruments and associated electronics for remote sensing satellites, antennas and communications electronics for communications satellites, telescopes and associated electronics for scientific satellites.

The platform or bus

Composed of any service subsystems required for supporting the general spacecraft operations including : solar arays, attitude and orbit control, propulsion, thermal control, structure, telemetry, telecommand, on-board data management.

Spacecraft mission operations generally consist in managing the spacecraft : more specifically, its two main parts, the payload and the platform for meeting given mission goals (e.g., accommoding customer's requests or scientific objectives).

Unmanned spacecraft are automatic systems that are teleoperated from the ground. These satellites operate at varying degrees of autonomy with the level of autonomy depending on several factors :

Type of orbit

For earth observation satellites stationed in low earth orbits (altitudes of 600-800km) the visibility from the ground center is limited to typically 10% of the mission duration, thus requiring some automatic operations on-board the satellite during the 90% of the orbit when control communications are interrupted.

Type of mission

Military communications satellites are far more autonomous than civilian ones.

Economic constraints

Commercial space communications organizations such as INTELSAT, INMARSAT or EUTELSAT are committed to reducing their operations costs, and consequently, there is motivation to automate the mission operations.

Considering the current state of the art, whatever the level of autonomy, most of the mission operations tasks are still performed on the ground. These tasks are, if we take the example of an earth observation satellite system, concentrated in two main ground entities :

- Mission Control Center (MCC)

- Spacecraft Control Center (SCC).

The first entity, the Mission Control Center, is primarily dedicated to Mission Planning, i.e., generating mission plans to be executed by the spacecraft through telecommands sent by the Spacecraft Control Center. Mission plans are generated in accordance with mission goals or customer's requests and in consideration of general mission conditions (e.g., spacecraft orbital position, seasonal condition), the current state of the spacecraft (provided by the SCC to the MCC on the basis of telemetry analysis), and, depending on the type of mission, the mission data reception (e.g., images for an earth observation satellite).

The second entity, the Spacecraft Control Center, is responsible for controlling the mission execution through telemetry parameter analyses. These parameters are regulary downlinked by the spacecraft. The SCC also commands and monitors the mission execution, according to the mission plan provided by the MCC and on the basis of predefined procedures. Thus, the main SCC functions are telemetry and telecommand processing.

Additional functions of the SCC :

Flight Dynamics

Compute spacecraft orbital positions, and thus determine appropriate orbital maneuver for orbital corrections.

Paper presented at the Mission Systems Panel of AGARD and the Consultant and Exchange Program of AGARD held in Madrid, Spain, on 6-7 November 1995; Châtillon, France, 9-10 November 1995; and NASA AMES Research Centre, California, USA, 16-17 November 1995 and published in LS-200.

Specific Software Packages

Perform specific monitoring tasks, e.g., on board electrical power balance between consumption and generation.

The architecture of the system described above is depicted in Figure 1/1.

All of the functions described above are performed by any spacecraft ground system, but they may be organized in different ways from one system to another. If we focus on Guidance and Control related functions, the functional analysis corresponding to such a system can be represented as that shown in Figure 1/2.

These functions may be performed on-ground (for most of the current space systems) or shared between on-board systems and a ground segment (for unmanned spacecraft with a high level of autonomy, or for manned spacecraft such as space shuttles or space stations).

These main components of any spacecraft operations are all candidates for partial or total automation, and thus, have the potential to benefit from applications of knowledge based systems.

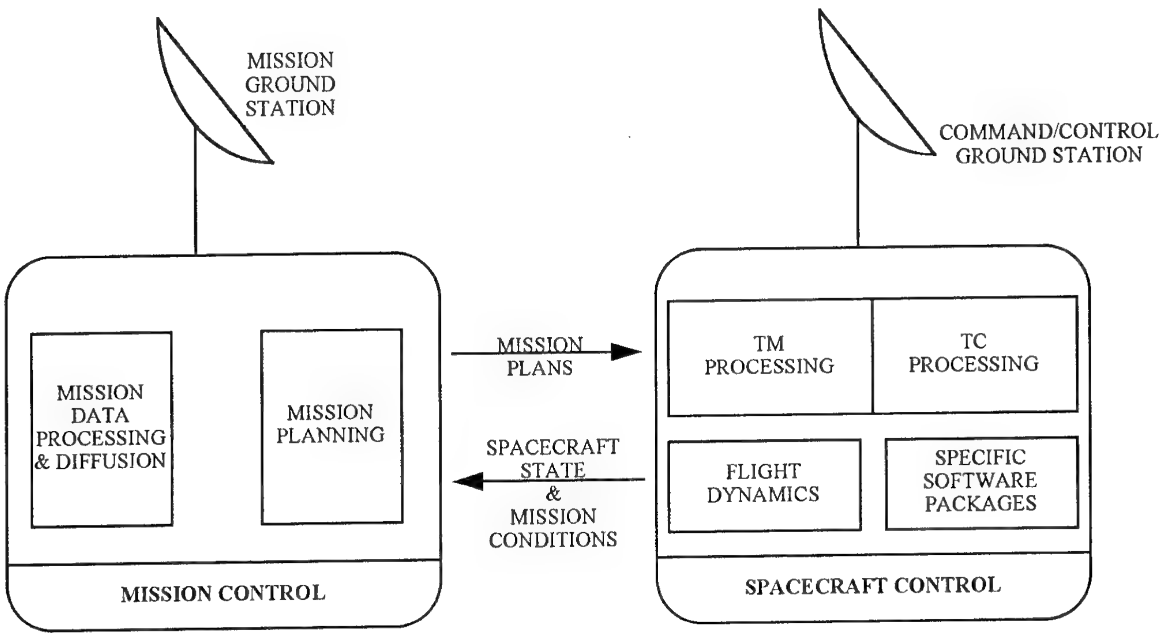


FIGURE 1/1 - GENERAL ARCHITECTURE OF A SPACECRAFT GROUND SEGMENT

2 - NEW CHALLENGES FOR SPACECRAFT OPERATIONS & KBS APPLICATIONS

Current trends in spacecraft mission operations (ref 4) are requiring a breakthrough for what concerns philosophy, organization and support tools, in order to meet a sufficient level of safety, productivity and mission return and thus provide an appropriate context for KBS applications :

a. the human operator is facing spacecraft which are becoming more and more complex

b. space missions management complexity is increasing

c. space projects and missions duration is continuously increasing, thus creasing problems of expertise & experience capture

d. space systems require more and more flexibility and adaptability

e. space missions are generating huge amount of data, from which essential informations are very difficult to extract.

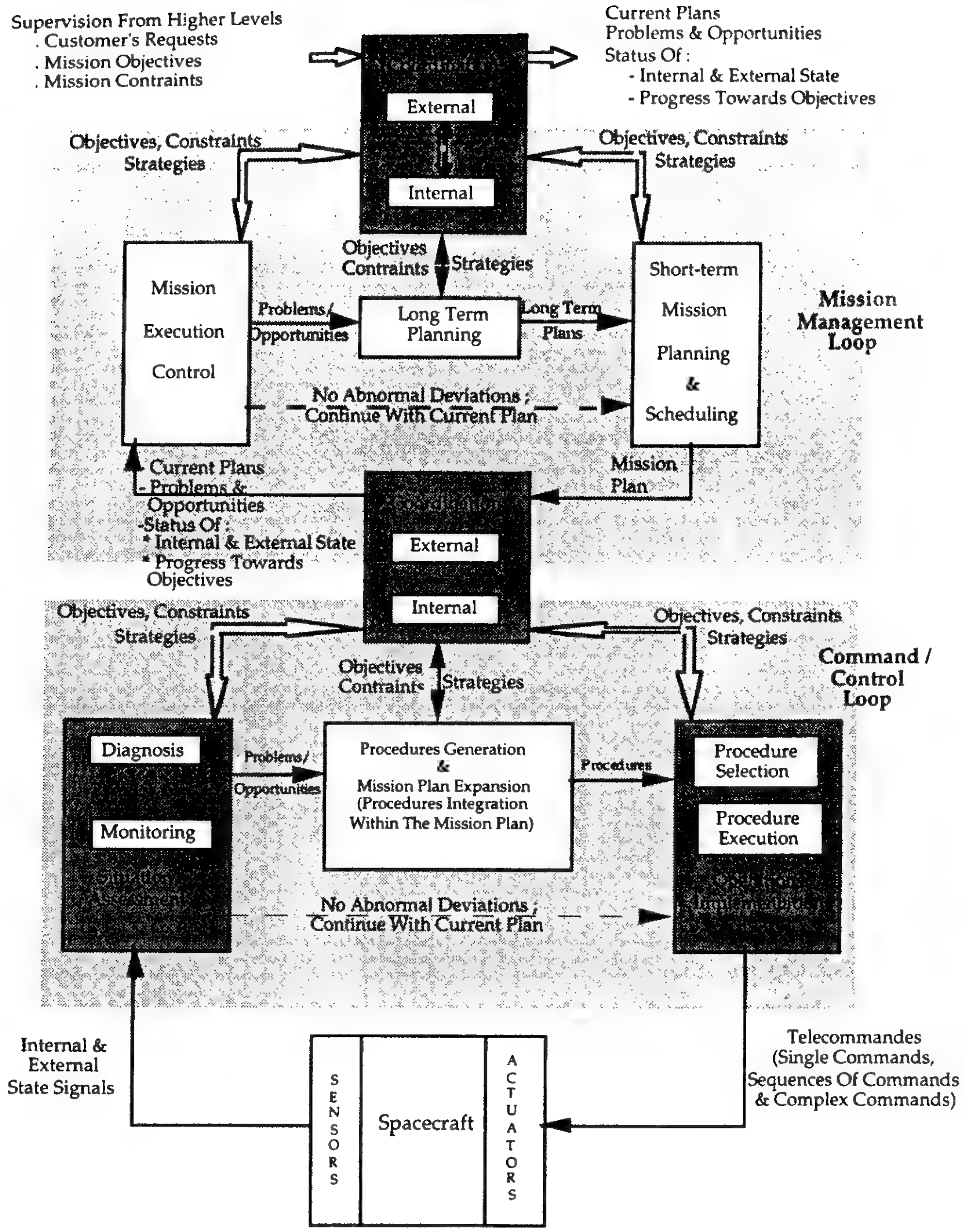


FIGURE 1/2 - FUNCTIONAL STRUCTURE OF SPACECRAFT MISSION OPERATIONS

Such a breakthrough is currently being implemented by the "informationalization" (ref.1) of space operations, i.e. a more efficient management of informations generated and used by space operations, and the automation of operations tasks, which used to be performed manually. This "informationalization" corresponds to the deployment of a complete set of tools, in addition to current operations and data processing systems. We proposed to call OPSWARE™ this new level in ground data systems. OPSWARE™ covers all the major mission operations tasks, as presented at figure 2/1

3 - SOME BRILLIANT ILLUSTRATIONS

During the past five years, numerous OPSWARE™ applications have been deployed, which address the various facets of OPSWARE™, as presented at figure 2/1.

Figure 3/1 gives a non-exhaustive list of some examples, developped by MATRA MARCONI SPACE.

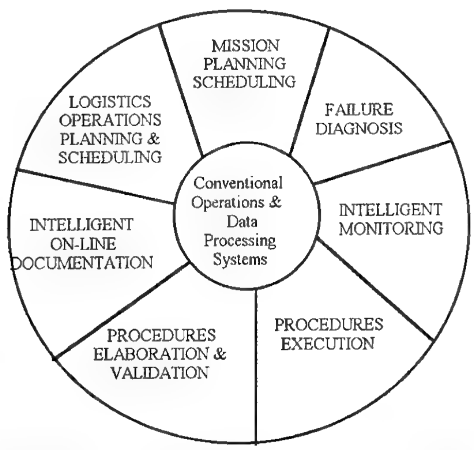


FIGURE 2/1 OPSWARETM : A COMPLETE SET OF TOOLS

OPSWARE™ FACET	CUSTOMER(S)	APPLICATION
MISSION PLANNING & SCHEDULING	ESA / ESTEC ESA / ESRIN	PLAN ERS FOR ERS1 &ERS2 EARTH OBSERVATION SATELLITES MISSION STRATEGY DEFINITION AND SIMULATION
LOGISTICS OPERATIONS PLANNING & SCHEDULING	MMS	- ARIANE 4 EQUIPEMENT BAY ASSEMBLY, INTEGRATION & VALIDATION - EARTH OBSERVATION INSTRUMENTS A.I.V.
INTELLIGENT ON-LINE DOCUMENTATION	MMS	COMMUNICATIONS SATELLITES OPERATIONS MANUAL
PROCEDURES ELABORATION & VALIDATION	CNES / MMS / ESA	POM POR TELECOM, HISPASAT & SOHO PROCEDURES ELABORATION
	ESA / ESTEC	PREVISE FOR MANNED FLIGHT PROCEDURES ELABORATION & VALIDATION (SPACELAB, COLUMBUS
	CNES	PROCSAT FOR SPOT EARTH OBSERVATION SATELLITE PROCEDURES ELABORATION & VALIDATION
	MMS	OPSAT FOR COMMUNICATIONS SATELLITES
PROCEDURES EXECUTION	ESA / ESOC	EXPERT OPERATOR ASSOCIATE FOR MARECS B2 SPACECRAFT OPERATORS ASSISTANCE
	ESA CNES	CREW SUPPORT SYSTEM FOR ASTRONAUTS ASSISTANCE & TRAINING
	MMS	OPS-EXECUTER FOR COMMUNICATIONS SATELLITES
INTELLIGENT MONITORING & DATA ANALYSIS	ARIANESPACE	ARIANEXPERT FOR ARIANE 4 POST - FLIGHT DATA ANALYSIS
	MMS	TELECOM2 & HISPASAT SATELLITES PERFORMANCES ANALYSIS
FAILURE DIAGNOSIS	CNES	TELECOM2 FAILURE DIAGNOSIS

FIGURE 3/1 - OPSWARE™ APPLICATIONS DEVELOPPED BY MATRA MARCONI SPACE

The following sections present some of these applications.

4 - PROCEDURES PREPARATION

A key task to be carried out during space mission preparation is the generation and validation of operation procedures. This is a complex and costly task, which concerns many entities in a project. This has motivated the development of several applications dealing with procedures preparation support.

The POM application has been developed by MMS in 1989-1990 to support the generation and maintenance of satellite ground control procedures, and to facilitate their use during operations thanks to a procedure browser (ref 5). POM is used operationally for the procedures of the Telecom 2, HISPASAT and SOHO spacecrafts. Savings that can be credited to POM during the procedure elaboration phase at MMS were estimated at 50%. Another fine result was the increase of procedure quality.

From the experience of the various procedures management tools developed in the last six years, MMS has derived OPSMAKER, a generic tool for procedure elaboration and verification. It has been applied to quite different types of missions, ranging from crew procedures (PREVISE prototype (ref 6)), payload data center procedures (PROCSU application), and satellite control procedures (PROCSAT developed for CNES, to support the preparation and verification of SPOT 4/HELIOS1 observations satellites operations procedures, and OPSAT for MMS telecom satellites operation procedures).

The basic functions provided by OPSMAKER procedures preparation applications are :

- a procedure editor which supports "assisted editing" (e.g. on-line access to system data) for more efficient procedures writing
- a procedures compiler, which generates an internal, formal representation of the procedures, detects syntactic errors and verifies the consistency of procedures with respect to the system database
- a procedures formater, which generates automatically a high-quality document (FOP)
- a procedures checker, which provides a rich set of verifications to speed up procedure developement : simple errors are detected early before starting detailed simulations.

These functions are detailed here below.

i. OPSMAKER provides a specialized editor for defining procedures and procedures books (FOPs)

   Procedures are written in a simple language, which is a straightforward normalization of the natural language usually used in operations (an example is visible in figure 4/1). They can thus be easily edited and maintained by operations engineers.

   The procedure is entered in a special from, with several fields dedicated to the body of the procedure and to other informations which can be attached to the procedure (author, type, preconditions,...)

   The procedure body editor is structured into columns which reflect the structure of the procedural instructions.

   An assisted editing environment is provided, with powerful assistance functions which make procedures preparation efficient : quick access to system data (TM, TC, TC blocks, ground system data), syntax driven editing, as well as a variety of search mechanisms are provided. For instance, it is possible to search for all procedures using a given step or TC, etc...

ii. From the procedure entered by the user, the OPSMAKER procedure compiler verifies the consistency of procedures with the system database, and automatically generates an internal representation in which all control structures (IF.. THEN.. ELSE.., CASE.., WHILE.., etc...) and elementary instructions (SEND TC..., CHECK TM..., ...), have been recognized and mapped into a data structure. This executable form will be the input for OPSEXECUTER (§6), but is also used locally in OPSMAKER for two complementary functionalities which are the procedure formater and the procedure verifier.

iii. The procedure formater allows to automatically generate a high quality document. It generates a command file for a standard deskstop publishing tool (e.g. FrameMaker). Data from the database are automatically inserted in the procedures (e.g. verification TM for a TC, list of TCs for a block) to build up the operators view. The procedure or a complete FOP can be formatted at a time.

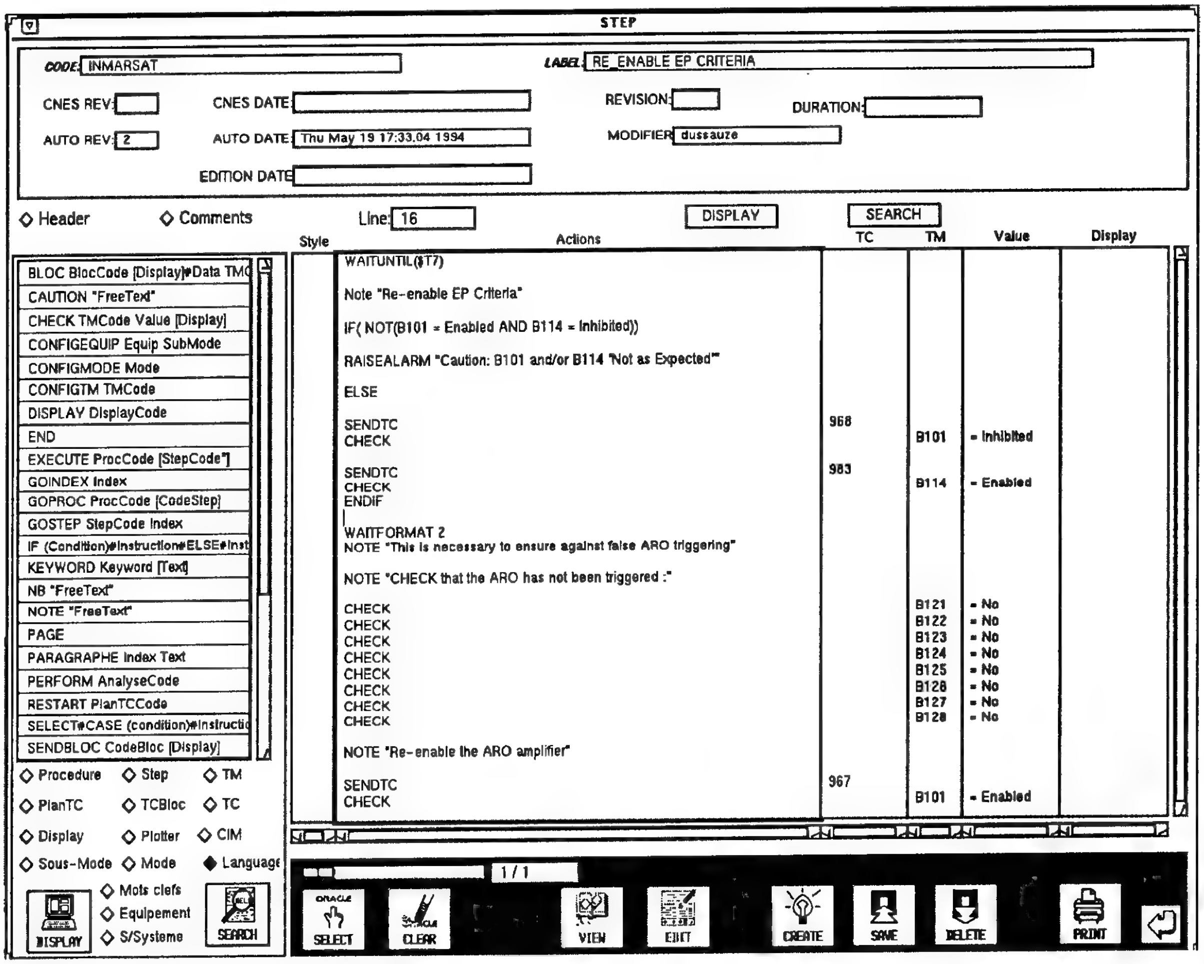


FIGURE 4/1 - A SCREEN OF THE OPSMAKER PROCEDURE EDITOR INTERFACE

Showing a part of an eclipse procedure written in the OPSMAKER language. Procedures are entered in a simple language and can easily be edited and maintained by operations engineers. They can specify command sequences, monitoring specifications, conditional paths, and reactions to events. The OPSMAKER procedure editing environment, with on-line access to system data and powerful search mechanisms, allows efficient procedure preparation

iv. The procedure verifier provides verification mechanisms ranging from simple "local" checks on the individual consistency of every statement, up to the "logical" verification of a procedure by simulating the effects of commands and checking operations constraints (e.g. TC and TC groups pre-validation checks).

These verification functions work on the basis of information stored in the TM/TC database. They allow to detect

errors or inconsistencies in procedures (e.g. missing or wrong commands), and thus to pre-validate the procedures before execution.

The general architecture of a typical OPSMAKER applications is shown in figure 4/2.

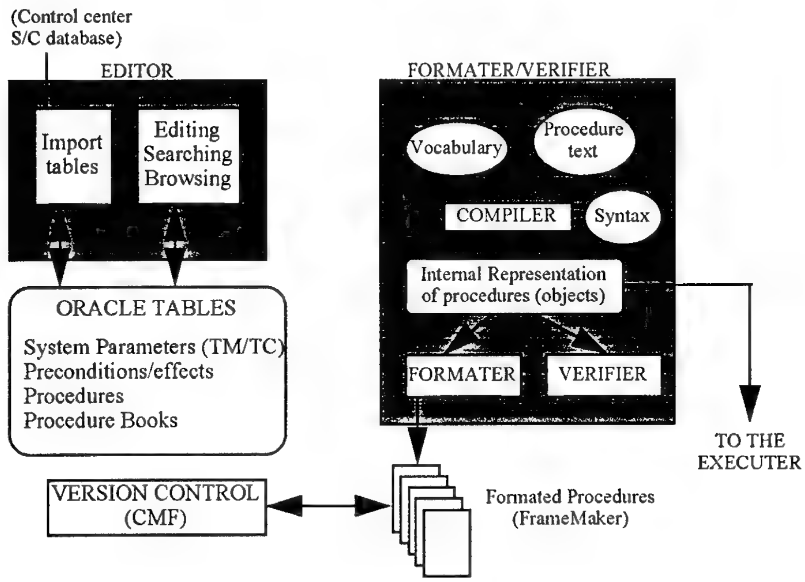
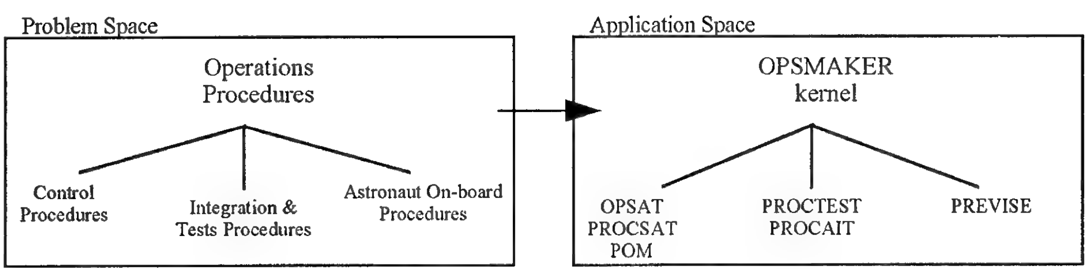


FIGURE 4/2 - OPSMAKER ARCHITECTURE

Once compiled by OPSMAKER, the procedures are ready for use by OPSEXECUTER. With OPSMAKER, they can also be formated into a FrameMaker document, and submitted to automated verification mechanisms.

In the PROCSAT and OPSAT applications of the OPSMAKER tool, procedures are saved in a relational database enabling fast search functions and safe team work : several instances of the editor can be opened at the same time (client server architecture). PROCSAT has been interfaced to a version control tool so that one can get back to a previous issue of the operations plan or print the changes from the previous issue.

Formalisation of procedures and modelling of actions facilitate team work by guarantying homogeneous procedures manuals. Everybody works at the same level of detail, with the same language. Maintenance of procedures is facilitated since information is never duplicated and powerful search functions are provided. The use of normalised language and a normalised presentation by the operations team is a factor for safer operations. Consistency checking of the operational data and the use of these data without possible corruption improves the consistency and quality of procedure manuals.



New applications of OPSMAKER to avionics test procedures (PROCTEST) and integration procedures (PROCAIT) are under development. This shall facilitate the exchange of procedures between the integration teams and the operations team.

5 - PLAN GENERATION

5.1 - Mission Planning

MMS has recently developped a tool for mission planning and scheduling, called TIMELINE, which was operational in April 1995. This tool is developped with two main perspectives :

- to assist the preparation of the timelines of the Launch and Early Orbit Phases (LEOP), taking into account ground stations visibilities, relative positioning of the spacecraft, the Earth and the Sun, as well as other constraints ;
- to provide a more general mission planning facility that will provide inputs to the OPSEXECUTER system.

The concept of timeline, as supported by the tool, is based upon three main types of entities : *activities, events, and resources.*

Activities may consist in operational procedures, or in other types of ground segment activities ; they are directly accessed from the operations database (ORACLE). The timeline is structured (i.e. split) into several *domains*, reflecting distinct areas of activity (e.g. platform management, payload management, ground segment activities,etc...). The definition of the domains is configurable by the user.

Events and resources can be defined directly within TIMELINE, using dedicated editors; They can also be imported from an external tool, using an import functionality : for instance events and resources related to orbitography can be automatically imported from a Flight Dynamics package.

Four types of planning constrains can be managed :

- precedence links between activities or between events and activities
- resources required by activities
- "logical constraints", e.g. system state conditions required at the beginning of an activity
- user-defined rules such as mutual exclusion between classes of activities,etc...

TIMELINE is designed to make the scheduling process as easy and flexible as possible, and provides facilities that allow to combine manual (interactive) scheduling and automated scheduling.

The events and resources availability profiles generated by the Flight Dynamics package are automatically imported, transformed into operational activities, and scheduled into a first timeline.

The user interface provides a graphical view of the timeline, with direct manipulation possibilities allowing to perform manual scheduling in an interactive way. Activities, events and resources, as well as links between activities can be simply placed or moved in the timeline using direct mouse operations.

More advanced functions will also be provided in next release :

- automated scheduling, taking into account the four types of constraints mentioned above
- Timeline verification against those constraints
- Timeline management functions, such as merging / splitting of sub plans.

TIMELINE generates two kinds of outputs :

- a printable file including a graphical view of the timeline
- a formal representation of the timeline (an ASCII file), which can be exchanged between several users of the TIMELINE tool, or transmitted to OPSEXECUTER for an automated execution.

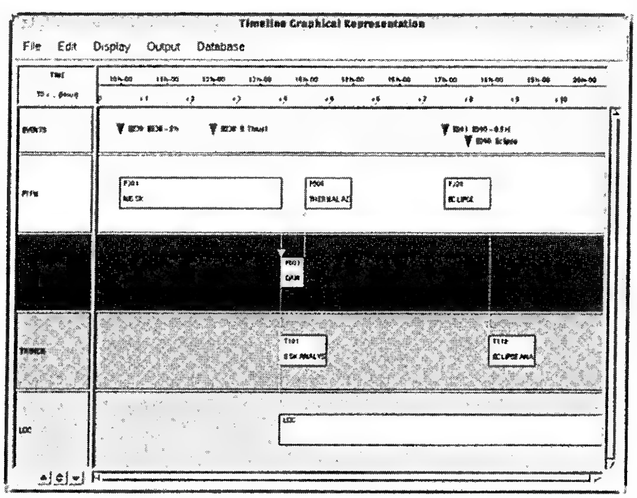


FIGURE 5/1 - A PART OF THE TIMELINE USER INTERFACE

showing some events, procedures from three domains, and precedence links between procedures.

5.2 - User Requests Scheduling

For some missions, the first step of the mission planning process deals with the scheduling of users requests ; these users requests concern the exploitation of the entire system (ground and space segments), and may thus result into on-board activities (e.g. on-board instruments activations, on board data management system,...) as well as ground activities (e.g. data processing, data dissemination).

Taking as an input a large set of pending requests, the objective of this first planning step is thus :

- to select a subset of these requests which can be handled during the next planning period,
- to map them into a set of operational (ground and board) activities,
- and to schedule those activities within that planning period.

The result of that process is a partial plan or timeline, which will have to be merged with other sub-plans (e.g. platform management plan) into a unique, consistent timeline. That latter part will be performed using the TIMELINE tool presented above (§5.1).

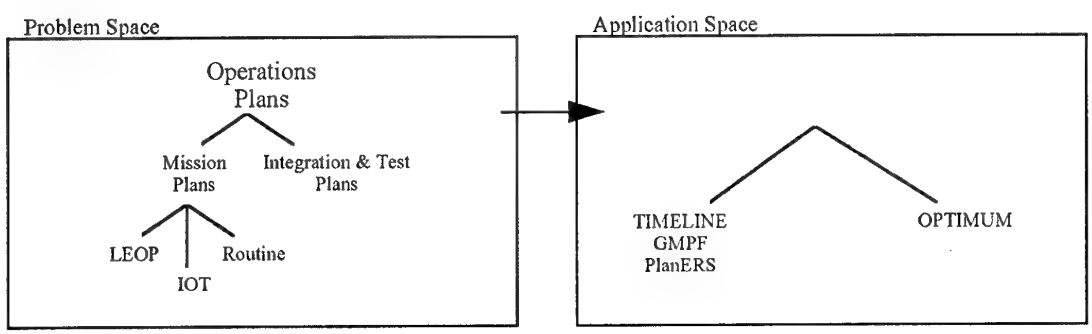
A fundamenal characteristics of this mission planning problem is its mission dependance : the constraints underlying the planning process, and theway users requests can be transformed into operational activities, are obviously dependent upon the type of spacecraft and mission.

The Generic Mission Planning Facility (GMPF) project, conducted by CRAY Systems and MMS for ESA/ESOC, has the objective of developing a generic mission planning "toolbox", which can be specialized for different types of missions. An object oriented approach was selected in oder to cope with the generic problem. This two years project, started in 1994, will thus lead in 1995 to a C++ library, compatible with the new ESA Mission Infrastructure (SCOS II) (ref 7), and providing :

- generic classes that represent the basic "data structures" suited to represent the planning problem (planning entities such as activities, events,..., planning constraints, planning strategies,...)
- basic algorithms (methods) covering the generic types of processing involved in the mission planning process.

MMS is in charge in this project to develop the GMPF generic library. By specializing whenever necessary the GMPF classes and methods, it will then be possible to develop, at low cost, a mission planning system suited to a particular mission.

5.3 - Other Operations Planning Systems



In addition to the tools presented above, MMS can rely on a wide experience on previous operations planning systems, and in particular :

- the plan ERS system (ref 8 & ref 9) for ERS observation satellite mission analysis : this mission planning system has been used (resp.at ESA/ESTEC and ESA/ESRIN) for performing several kinds of mission analyses for ERS1 and ERS2, in order to define mission planning strategies
- the OPTIMUM system (ref 10) : this generic project planning system has two operational applications at MMS, dedicated to the planning of Assembly, Integration and Verification activities
- internal developments of new algorithmic techniques for earth observation satellite mission planning, these techniques allow to deal very

efficiently with the combinatorial complexity of the planning problem, and to optimize as much as possible the use of ground and board resources.

6 - OPERATIONS AUTOMATION

The automation of satellite operations is a key for reducing exploitation costs and increasing mission safety. For these reasons, MMS has been working for several years on automated execution of timelines and procedures, and is now developing an operational system called OPSEXECUTER.

OPSEXECUTER is being developed starting from the experience acquired by MMS in the previous Expert Operator Associate (EOA) project, conducted with CRI for ESA/ESOC from 1988 to 1992, and experimented in 1993 (ref 11). In this project, a prototype for automated operations execution was developed, interfaced to the ESOC Multi Satellite Support Systems (MSSS), and experimented with MARECS B2 spacecraft analysis. An another key input for OPSEXECUTER is the OPSMAKER procedure preparation tool (§4).

Functionally, the objectives of OPSEXECUTER are identical to those of EOA, but the software is built in the perspective of a fully operational system, and inter operable with the OPSMAKER tool and with the TIMELINE tool for operations planning / scheduling.

OPSEXECUTER has been operational since May 1995 ; its first version is connected to the MMS satellite control facilities : the "Customer Support Center".

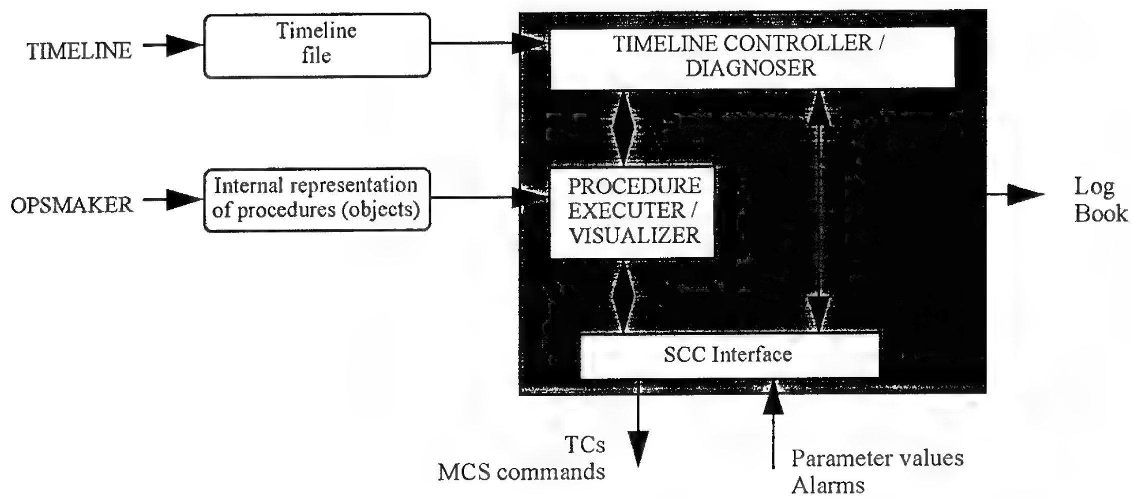


FIGURE 6.1 - OPSEXECUTER ARCHITECTURE

The tool allows to fully automate the execution of sequence of procedures specified in a timeline. It is easily adaptable to various mission control centers, and check-out systems.

OPSEXECUTER includes the following modules :

- a timeline controller/diagnoser composed of four main functions
  - managing the procedures timeline (activation of procedures based on the timeline),
  - monitoring alarms
  - performing a first-level diagnosis in case of contingency
  - providing a high level view of the timeline under execution
- a procedure executer/visualizer automating the execution of a procedure activated by the timeline controller. The procedure is shown on the user interface in a similar layout as the paper document, with the current instruction highlighted. Thanks to the OPSMAKER operations language, the execution is fully observable. The user has the possibility to interrupt at any time the procedure under execution, for instance to take manual control, or to continue the execution in a step-by-step manner.
- a generic interface to the core mission center, that allows :

- sending requests for sending spacecraft commands and/or activating ground segment functions,
- sending requests for parameter values
- receiving events (alarms, command execution acknowledge,etc...)

A logbook of events (procedures activated, commands sent) is maintained and saved into a file. In a next version, each command execution report and command execution date shall be printed in a dedicated column of the executer in front of the command to provide more efficient reporting. Skipped commands will be instatly visible.

OPSEXECUTER is designed as a complementary module to the core functionalities of the spacecraft control center (parameter processing and monitoring, spacecraft and ground segment commanding). It is developed as a portable C++ software.

OPSEXECUTER is fully integrated with the OPSMAKER system for procedures preparation. There is a unique representation for a procedure, which is simultaneously printable into a document used for conventional (i.e. manual) execution, and usable for an automated execution by OPSEXECUTER : thus, there is no need for recoding procedures, and no duplication of information. This simplifies validation and maintenance of procedures. Furthermore, all the information that will be used by the executer during on-line execution of the procedure is expressed in the OPSMAKER language : there is no hidden instruction or obscure mechanism involved at execution time.

OPSEXECUTER provides full observability and controllability during procedure and timeline execution. It shall relieve operators from low level monitoring during nominal operations, and shall also facilitate investigations and manual control.

7 - LESSONS LEARNED FROM THE DEPLOYEMENT OF OPSWARE™ APPLICATIONS

As briefly described in the previous sections, MATRA MARCONI SPACE has delivered several OPSWARE™ applications, thus giving interesting user's feedback.

It is possible to extract some rules from this feedback:

a. OPSWARE™ systems can be implemented as extensions to existing space operations and data processing systems. Some implementations are rather straight forward (intelligent on-line documentation, failure diagnosis, procedures elaboration & validation,...). Some others may require some adaptations of the existing operations & data processing infrastructure for enabling it to integrate OPSWARE™ modules (operations execution, mission planning & scheduling,...). But the great point is that OPSWARE™ systems can be added to existing infrastructure. The evolution in the way operations tasks are performed, associated to OPSWARE™ implementation, is probably more significant (see b).

   Furthermore, some current trends is ground data systems for spacecraft control (e.g. distributed architecture, network of workstations,...) are propitious to OPSWARE™ deployment.

b. User's motivation is crucial to OPSWARE™ implementation success, and this comes from two facts. First, a basic requirement for OPSWARE™ systems is a perfect human-machine synergy, whatever the concerned task. Second, if operations infrastructure is not necessarily deeply affected by OPSWARE™ deployment, it is completely different for what concerns operations tacks themselves ; they used be performed manually, and are now performed by an "integrated human-machine intelligence" (ref 12). User's motivation is thus mandatory !.

c. A specific lifecycle is required for this type of systems. A fisrt delivery to the user is required as soon as possible during the project, which will give a user's feedback very early, and so will guarantee the matching between system specified functionalities and the actual needs of the user. More generally, an incremental development is required to rapidly implement changes, which may be frequent for this kind of tools. The programming technologies which are used in OPSWARE™ systems (object oriented programming, artificial intelligence, hypertext, interface builders,...) perfectly support such a lifecycle (ref 2).

d. A more general lesson, which has been illustrated in this paper, is that OPSWARE™ is operational now, and has already deeply modified space operations philosophy, and has demonstrated a R.O.I. (return on Investment) (ref 13).

8 - MAJOR STAKES FOR THE COMING DECADE & CONCLUSIONS

The deployment of these knowledge-based operations support tools we call OPSWARE™ is already a major fact in space operations, and OPSWARE™ stakes for the coming decade are paramount. They can be summarized in "Reengineering Space Operations" (ref 2). What are they ?.

a. Optimizing Space Missions Performances, thanks to powerful and flexible planning & scheduling

systems optimizing the mission end products, smarter performance analysis tools and operations support tools allowing a faster operator's reaction.

b. Reducing Space Operations Cost & Duration, thanks to operations automation, and an optimized resources management. This objective is more or less difficult to meet, depending on the nature of the space project ; the problem is completely different between a communications satellites project and a space station program, even is the objective is critical for both.

c. Enhancing Space Operations Safety, thanks to a better human machine synergy, and to operations automation providing exhaustivity and systematization.

   The goal here is to reduce the number of operations errors, which still occur too much frequently in space projects, and which may have critical consequences.

d. Capturing Design Knowledge and In Orbit Experience, thanks to the implementation of a computer integrated technical memory, and to expert tools for operations support.

These stakes are critical for the success of future space missions, and concern all the international space communities.

OPSWARE™ systems is a key to them. They can play a major role in making space operations faster, better and cheaper!.

REFERENCES

1. **STAN DAVIS & BILL DAVIDSON 1991**

   2020 VISION, Transforming your business today to succeed in tomorrow's economy.

   A fireside book. Published by Simon & Schister.

2. **INFORMATION WEEK, Bonus issue,**

   May 5 1992

   How you fit in... Reengineering... and how you don't.

3. **AGARD ADVISORY REPORT N°325, January 1995**

   Knowledge Based Guidance and Control Functions.

4. **JM. DARROY, June 1990**

   Knowledge Based Systems for Spacecraft Control, Keynote adress

   1st International Symposium on Ground Data Systems for Spacecraft Control Darmstadt, Germany.

5. **JY. MARSON, P. GREMILLON**

   An hypermedia tool for operational procedure creation and utilisation, First International Symposium on Ground Data Systems for Spacecraft Control, Darmstadt, Germany, June 1990.

6. **A. DE SAINT VINCENT, F. LECOUAT, G. LEONIS, F. ALLARD**

   PREVISE : a KBS to Support the Preparation and Verification of Crew Procedures, 4th ESA Workshop on Artificial Intelligence and Knowledge Based Systems for Space, Noordwijk, The Netherlands, May 1993.

7. **JF. KAUFELER, NC. HEAD**

   SCOS II : ESA's new generation of Mission Control Systems, Proc. Space Ops 92, Pasadena, USA, 1992.

8. **J. FUCHS, A. GASQUET, J. MULLER, P. VIAU**

   The Expert Planning System Prototype : Experience and users Feedback, ESA Workshop on Artificial Intelligence and Knowledge Based Systems for Space, Noordwijk, The Netherlands, May 1991.

9. **A. GASQUET, Y. PARROD, A. DE SAINT VINCENT**

   Towards a New Generation of Mission Planning Systems : Flexibility & Performance, Proc. Space Ops 94, Greenbelt, MD USA, November 1994.

10. **M. ARENTOFT, J. FUCHS, Y. PARROD, A.GASQUET et al,...**

    OPTIMUM-AIV, A Planning and Scheduling System for Spacecraft AIV, ESA Workshop on Artificial Intelligence and Knowledge Based Systems for Space, Noordwijk, The Netherlands, May 1991.

11. **F. LECOUAT, M. NIELSEN, K. GRUE**

    Knowledge Based Assistance for Generating and Executing Space Operations Procedures, ESA Workshop on Artificial Intelligence and Knowledge Based Systems for Space, Noordwijk, The Netherlands, May 1991.

12. **JM. DARROY, September 1990**

    Integrated Human Machine Intelligence in Aerospace.

    Conference introduction by the conference chairman

    3rd International Conference on "Human Machine Interaction & Artificial Intelligence in Aeronautics & Space" TOULOUSE, FRANCE.

13. **JM. DARROY, F. LECOUAT, JM. BRENOT, A. DE SAINT VINCENT,**

    OPSWARE : A new generation of software tools for making space operations faster, better and cheaper, 44th Congres of the International Astronautical Federation, Graz, Autria, October 1993.

Design Principles and Algorithms for Automated Air Traffic Management

Heinz Erzberger
NASA Ames Research Center
MS 210-9, Moffett Federal Airfield
CA 94035-1000
USA

Abstract
This paper presents design principles and algorithm for building a real time scheduler. The primary objective of the scheduler is to assign arrival aircraft to a favorable landing runway and schedule them to land at times that minimize delays. A further objective of the scheduler is to allocate delays between high altitude airspace for from the airport and low altitude airspace near the airport. A method of delay allocation is described that minimizes the average operating cost in the presence of errors in controlling aircraft to a specified landing time.

INTRODUCTION

The urgent need for increasing the efficiency of the air traffic management process has led to intense efforts in designing automation systems for air traffic control. The efforts have been dominated by two major technical challenges: designing a trajectory synthesizer/estimator and a real time scheduler. The design of the trajectory synthesizer/estimator, though technically complex, can be accomplished by the application of well established methods for navigation, guidance and control of aircraft [1], [2]. In contrast, the design of the real time scheduler has no technical precedence to build upon and has been found to require a unique blend of expert knowledge of air traffic control and analytical procedures. It is therefore an especially appropriate subject for this lecture series.

The scheduler described herein is incorporated in the Center TRACON Automation System (CTAS) which is being developed jointly by NASA Ames Research Center and the Federal Aviation Administration. The automation tools in CTAS consist of the Traffic Management Advisor, (TMA) the Descent Advisor (DA) and the Final Approach Spacing Tool (FAST) [3]-[5]. These tools generate advisories that assist controllers in handling aircraft from about 40 minutes of flying time to an airport, until they reach the final approach fix. While the design of CTAS is not yet complete, several of its tools have undergone extensive real time simulation tests as well as field evaluation at the Denver and Dallas/Fort Worth airports [6]-[8].

ROUTE STRUCTURE AND SCHEDULING CONSTRAINTS

The basic objective of the scheduler in air traffic control automation is to match traffic demand and airport capacity while minimizing delays. As we shall see, this concise and straight forward sounding objective gives rise to a surprisingly complex algorithmic design problem when all necessary operational constraints are considered. In this chapter we present an outline of the solution to this problem.

The dynamic nature of air traffic flow requires that the scheduler be designed to operate as a real time process which is defined in the following way. The scheduler must generate an updated schedule for the set of aircraft to be scheduled both periodically and in response to aperiodic events. The length of the periodic cycle is related to the basic radar update rate, which is 10-12 seconds in length. In Center airspace, experience has shown the scheduler update cycle must be a small multiple of the radar update cycle, or in the range of 30-60 seconds. Aperiodic events requiring an immediate update of the schedule are primarily due to controller inputs such as a

Paper presented at the Mission Systems Panel of AGARD and the Consultant and Exchange Program of AGARD held in Madrid, Spain, on 6-7 November 1995; Châtillon, France, 9-10 November 1995; and NASA AMES Research Centre, California, USA, 16-17 November 1995 and published in LS-200.

change in airport configuration, a change in airport landing capacity, etc. While controllers prefer a nearly instantaneous response of the scheduler to such inputs, in practice a response within 10-15 seconds has been found to be acceptable and qualifies as real time performance.

The objective of minimizing delays implies that mathematical optimization must be performed by the scheduler in real time. However, it is recognized that an algorithmic solution of the full scheduling optimization problem with all important constraints included is infeasible and probably impossible. In light of this situation, numerous studies have been done to synthesize practical algorithms that combine both adequate scheduling efficiency and short computation times, so as to maintain real time performance.

The scheduling principle underlying all practical real time scheduling algorithms that have so far been developed is referred to as first-come-first-served (FCFS) [3], [9]. In general, this principle generates "fair" schedules when delays must be absorbed. It is also known to be an optimum schedule for a simple constraint condition and performance criterion. In the discussions to follow, this principle is therefore the starting point for important aspects of the scheduler design. A precise definition of FCFS in the context of air traffic scheduling will be given later.

In addition to the FCFS principle, the scheduling problem is characterized by numerous constraints. The complexity of the scheduling algorithm that remains true to the FCFS principle is greatly increased by the presence of these constraints, as will be seen when the algorithm is derived.

Nevertheless the scheduling algorithm described herein generates a feasible FCFS schedule without computationally lengthy iterative procedures, thereby achieving the precondition for real time operation.

Airspace and Route Structure

In order to serve the changing needs of airlines and air traffic control, the airspace and route structure surrounding a large airport have evolved into increasingly complex forms. Here we describe only those features that relate directly to the design of the scheduler.

For the purpose of the scheduler design, the arrival airspace is divided into Center and TRACON regions. Whereas the Center region has an irregular outer boundary, the TRACON region is a roughly circular area about 35 n.m. in radius and is completely surrounded by the Center airspace. Certain way points located on the boundary between the two regions are referred to as meter gates. During moderate and heavy traffic conditions when delays are expected, traffic is funneled through these gates as a means of controlling or metering the flow rate into the terminal area. In most terminal areas, arrival routes are merged at four gates corresponding to the primary arrival directions. An exception is the terminal area for the new Denver International Airport, where eight primary gates, grouped into four closely spaced pairs, are in use.

Traffic flowing to each gate is often segregated into two independent streams by separating each stream vertically by at least 2000 feet in altitude at the gate crossing point. This is done so as to permit the two primary aircraft types, jets and turboprops, which have significantly different airspeed ranges, to cross the gates independently, thereby avoiding conflicts due to overtakes near the gates.

From each gate, routes are defined in the TRACON airspace that lead to all possible landing runways for each independent stream. For the design of the scheduler, the exact horizontal path of the routes is not important; only the nominal flying times from each gate to all landing runways must be provided as input.

Figure 1 illustrates the concepts of airspace structure, arrival routes, meter gates and stream types as has been described above.

Scheduling Constraints: In-Trail Distance Separations

Scheduling constraints can be broadly classified into two types: In trail distance separation constraints and sequence order constraints. Here the former is discussed.

In the design of the scheduling algorithm, the in-trail constraints play an especially important role because they determine the capacity of an airport and hence the maximum landing rate. The scheduling algorithm must be designed to meet in-trail constraints both at the meter gates and on the final approach paths.

At the meter gates in-trail separations may be specified by separate parameters for each independent traffic stream in order to provide flexible control of the total flow rate into the TRACON. However, in the absence of flow rate control at the meter gates, safety considerations require the minimum in-trail distance separations to be not less than 5 n.m. Since scheduling is done in the time domain all distances must be converted into equivalent time separations. In general, the conversion first determines the ground speed from estimates of airspeed and wind speed and then applies the following relation:

$$T_{it} = D_{it} / V_g \tag{1}$$

where:

D_{it} = specified in-trail distance separation

V_g = estimated ground speed of trailing aircraft

T_{it} = time separation of trailing aircraft from leading aircraft when leading aircraft is at time control point

This relation not withstanding, a fixed value of one minute for the minimum time separation has been found to be adequate and is used throughout this paper.

On the final approach path the minimum in-trail distance separations are a function of both aircraft weight class and landing order as determined by the FAA's wake vortex safety rules. Table 1 gives the values in matrix format.

The table also gives examples of aircraft models falling in the different weight categories. A separate classification exists for "small," aircraft having landing weights below 12,500 lbs. Such aircraft do not contribute a large fraction of traffic at hub airports and therefore are not included. They could be included in Table 1 by adding a fourth row and column.

As before, the distance separation in Table 1 must be converted to equivalent time separation for use by the scheduler. The conversion process is complex and is given here only in outline. Generally it involves modeling the airspeed profile of each type of aircraft and the wind speed on final approach and then integrating the equations of motion along the final approach path. The result of this process is the time separation matrix given in Table 2 for the case of zero wind.

Scheduling Constraints: Sequence Order and the Definition of First-Come-First-Served

A sequence order constraint specifies the order with respect to time that a group of aircraft must be scheduled to cross a time control point. The runway threshold and the meter gates are the points where sequence constraints are frequently enforced. They provide an essential mechanism for achieving scheduling efficiency, scheduling fairness and controller preference. The sequence order that often meets the requirements of all three of these objectives simultaneously is the First-Come-First-Served (FCFS) order. It therefore plays the role of the standard or canonical sequence order against which all other orders are referenced.

Let $\{ETA(i)\}_N$ be the set of estimated times of arrival for the set of N aircraft $\{A_i\}_N$ at the runway threshold, where the A_i are the aircraft identifiers. Then the FCFS order at this point is the time-ordered list of this set of ETA's arranged in a vertical column, as shown in Table 3.

By convention and for economy of notation the earliest ETA at the bottom of the list is associated with aircraft A_1, while the latest ETA at the top is associated with aircraft A_N.

Table 1: Minimum distance separation matrix for pairs of aircraft simultaneously on final approach path, n.m.

	Trailing Heavy Jet	Trailing Large Jet	Trailing Large Turboprop
Lead Aircraft Heavy Jet (747, DC-10)	4.0	5.0	5.0
Lead Aircraft Large Jet (MD 80, 737)	2.5	2.5	2.5
Lead Aircraft Large Turboprop (AT 42, King Air)	2.5	2.5	2.5

Table 2: Minimum time separation matrix for pairs of aircraft on final approach, seconds

	Trailing Heavy Jet	Trailing Large Jet	Trailing Large Turboprop
Leading Heavy Jet	113	135	170
Leading Large Jet	89	89	110
Leading Large Turboprop	83	83	94

Table 3: FCFS Ordered List of ETA's

$ETA \quad (A_N) \quad (latest)$

$\vdots$

$ETA \quad (A_4)$

$ETA \quad (A_3)$

$ETA \quad (A_2)$

$ETA \quad (A_1)$

Table 4: Illustration of Position-Shifted Sequence Order relative to FCFS order

<u>FCFS</u> <u>Postiton Shifted Order</u>

$ETA\,(A_N) \longrightarrow A_N$

$\vdots$

$ETA\,(A_4) \longrightarrow A_4$

$ETA\,(A_3)$ $\quad A_2$

$ETA\,(A_2)$ $\quad A_3$

$ETA\,(A_1) \longrightarrow A_1$

A sequence list of aircraft not in FCFS order is said to be position-shifted. A position-shifted order can be displayed graphically by placing the FCFS order list of ETA's and the position-shifted list of aircraft identifiers adjacent to each other and then connecting corresponding ETA's and aircraft identifiers with lines as shown in Table 4.

The crossed lines identify the aircraft that are position-shifted. In Table 4, A_2 and A_3 are position-shifted by one, meaning that an order reversal of these two adjacent aircraft returns them to FCFS order. Higher order position-shifts would appear as multiple line crossings. If advancing an aircraft by k slots relative to FCFS is defined as a positive position shift of k and delaying it by l slots is defined as a negative position-shift of l, then it can be shown that the algebraic sum of all position shifts of an arbitrarily position-shifted sequence order is zero.

The basic sequence order constraints for which the scheduling algorithm will be derived consist of FCFS order at the runway and FCFS order for each independent stream at each meter gate.

The algorithm can easily be adapted to accommodate position-shifted sequence order at the runway or the meter gates. Position shifting is a technique for reducing delays by optimizing the landing sequence and will be discussed later.

Recently, Brinton developed an algorithm for sequence and runway assignment optimization using a variant of binary enumeration and branch and bound technique [10].

DESCRIPTION OF THE BASIC ALGORITHM

In this section the basic algorithm that generates schedules to the runway threshold while obeying FCFS sequence constraints at both meter gates and runways is described. The algorithm builds the schedule by a non-iterative constructive procedure that translates directly into a rapidly executing software program. While the algorithm packs aircraft as tightly together as the constraints permit, it does not optimize any specific performance functions. If sufficient computing power and time are available, the schedule generated by the basic algorithm, can, however, serve as the initial schedule for iterative algorithms, such as described in [10], that reduce delays by optimizing the landing sequence and runway allocations. The next chapter describes an optimization approach that works within real time constraints.

It is assumed that the schedulable aircraft are in Center airspace and some distance away from the meter gate. The basic input to the scheduler is the set of estimated times of arrival of all schedulable aircraft, computed to the appropriate meter gates. This set, designated by $\{ETA_{FF}\}$ is provided by the trajectory synthesizer algorithm. For the sake of simplicity but without loss of generality, the derivation is given for the case of two meter gates, A and B, and one runway. Aircraft assigned to these gates have associated identifiers $\{A_i\}_M$ and $\{B_j\}_N$, respectively.

Thus M+N are the total number of aircraft to be scheduled. Let $\{T_t(A_i)\}_M$ and $\{T_t(B_j)\}_N$ be the set of TRACON transition times. They specify the nominal time intervals required for aircraft to fly from their respective meter gates, A or B, to the runway threshold. Therefore, the estimated time of arrival of aircraft A_i at the threshold can be written as $ETA(A_i) = ETA_{FF}(A_i) + T_t(A_i)$, and similarly for aircraft B_j. The set of transition times are input quantities also generated by the trajectory synthesis algorithm.

A series of time lines will be used to illustrate various steps in the development of the scheduling algorithm. Each figure in the series consists of several vertical time lines arranged side by side representing a geographic scheduling point, either a meter gate or a runway. The transformation and procedures described in the various steps are

represented graphically by lines connecting objects on adjacent time lines. The objects are generally the aircraft to be scheduled. By studying the figures in sequence the reader can follow a specific scheduling problem from beginning to end.

Step 1: Apply in-trail separation constraints at meter gates

Let the set of scheduled times of arrival at the meter gates with in-trail constraints T_{it} be $\{STA_{FFit}\}$. Generate the $STA_{FFit}'s$ sequentially at each meter gate starting with the earliest to arrive aircraft A_1 and B_1 at gates A and B, respectively:

$$\left.\begin{array}{l} STA_{FFit}(A_1)=ETA_{FF}(A_1) \\ STA_{FFit}(A_2)=Greater\,of\left\{\begin{array}{l} ETA_{FF}(A_2) \\ STA_{FFit}(A_1)+T_{it} \end{array}\right\} \\ \quad\vdots \qquad\qquad \vdots \\ STA_{FFit}(A_M)=Greater\,of\left\{\begin{array}{l} ETA_{FF}(A_M) \\ STA_{FFit}(A_{M-1}) \end{array}\right\} \end{array}\right\} \quad (2)$$

and similarly for gate B aircraft. For generality the T_{it} parameter should be considered a function of the meter gate and stream type. This step is illustrated in Figure 2a.

Step 2: Determining the Runway Threshold Landing Order

As previously stated, the overall objective is to generate a FCFS order at the runway. However, when in-trail constraints are present at the meter gates, such as those described in step 1, the definition of FCFS at the runway becomes ambiguous. The ambiguity is removed by choosing the $STA_{FFit}'s$ rather than the $ETA_{FF}'s$ when establishing the FCFS order at the runway. Simulation and analysis have shown this choice produces both a fairer schedule overall as well as one that is slightly more efficient than a schedule that ignores the meter gate constraints.

The process begins by propagating the $STA_{FFit}'s$ forward in time from the gates to the runway by using the TRACON transition times. If $RTA(A_i)$ and $RTA(B_j)$ designate the runway times of arrival of aircraft $A_i\,and\,B_i$, then:

$$RTA(A_i)=STA_{FFit}(A_i)+T_t(A_i) \quad (3)$$
$$RTA(B_j)=STA_{FFit}(B_j)+T_t(B_j) \quad (4)$$

Repeating this for all schedulable aircraft results in the two sets:

$$\{\ RTA(A_i)\ \}_M,\{\ RTA(B_j)\}_N \quad (5)$$

The times in these sets represent the earliest possible landing times of the schedulable aircraft when in-trail constraints at the gates are included but in-trail constraints on final approach are ignored.

Before the FCFS landing order is determined from the sets of RTA's, an order rectifying procedure must first be performed, for the following reason. Because different aircraft types can have substantially different TRACON transition times, the RTA's in equation (5) are not necessarily in FCFS order. That is, the RTA order can become position shifted relative to the STA_{FFit} order for aircraft passing through the same gate. The occurrence of overtakes between aircraft in the same stream class flying from the same gate to the same runway is generally not acceptable to controllers and must be excluded by the scheduling algorithm. It is necessary to check each set of RTA's for position shifted sequences. If such sequences are found, the T_t of each overtaking aircraft is increased by the smallest time increment that modifies the RTA's so as to restore them to FCFS ordered sequences. It is now assumed that the RTA's in equation (3) have already been rectified in this manner and are therefore in FCFS order.

The runway landing order list, $\{C_p\}_{M+N}$, is now obtained by merging the two sets of RTA's into a FCFS time ordered sequence list:

FCFS landing order list:

$$\{C_p\}_{M+N}=\{A_{ik},B_{jl}\}_{M+N} \quad (6)$$

where the second indices k,l indicate the landing order. The indices satisfy the

$$
\left.\begin{array}{l}
STA(C_1)=RTA(C_1) \\
\\
STA(C_2)=Greater\ of\begin{Bmatrix} RTA(C_2) \\ STA(C_1)+T_{it}(C_1,C_2) \end{Bmatrix} \\
\vdots \\
STA(C_{M+N})=Greater\ of\begin{Bmatrix} RTA(C_{M+N}) \\ STA(C_{M+N-1})+T_{ij}(C_{M+N-1},C_{M+N}) \end{Bmatrix}
\end{array}\right\} \quad (7)
$$

inequalities $k \geq i,\ l \geq j$ over their range of values. Figure 2b illustrates the merging and ordering process. Note that no lines connecting gate sequences and landing sequences cross, as required by the overtake exclusion.

Step (3): Computing scheduled times of arrival at the runway threshold

In this step the time separations between the unconstrained runway times, the RTA's, are stretched, when necessary, to conform to the minimum time separation matrix given in Table 2. This yields the scheduled times of arrival, the STA's at the runway threshold.

The process involves inserting the appropriately chosen minimum time separation, T_{it}, from Table 2, between pairs of aircraft starting with the first aircraft in the known landing order, and terminating with the last. This process can be written symbolically as shown in equation set (7).

If the RTA's are closely bunched, thus requiring the $T_{it}'s$ to be inserted for a portion of the schedulable aircraft, the sequential character of equation (7) can propagate a delay ripple for successive aircraft in the landing order. The delay ripple terminates when a sufficiently large time gap occurs between successive RTA's. The delay $d(C_p)$ generated by equation (7) for the p^{th} aircraft in the landing order can be written as:

$$d(C_p)= STA(C_p) - STA_{FFit}(C_p) \quad (8)$$

In addition to the scheduling constraints already described, two other types of scheduling constraints referred to as blocked intervals and reserved time slots on the runway must also be handled by the scheduling algorithm. They are specified time intervals and virtual aircraft landing times that the scheduling algorithm must avoid when generating the STA's for the list of schedulable aircraft. The logic in equation (7) can be extended in a straight forward way to handle these constraints.

The processes described in this step are illustrated by the example in Figure 2c. A blocked time interval has been included as a constraint.

Step 4: Development of Delay Distribution Function

Whenever the total flow rate to a runway exceeds a certain maximum rate for a significant period of time, the separation constraints imposed by equation (7) will generate large delays. When that occurs it is said that the rate exceeds the runway capacity. Up to this point in the development of the algorithm, all delays no matter how large, would be absorbed between the meter gates and the runway. However, a group of aircraft in sequence, each with delays of several minutes to absorb in the TRACON airspace, creates excessive workload for TRACON controllers and can produce potentially

$$STA_{FF}(B_1) = STA(B_1) - T_t = ETA_{FF}(B_1) \tag{12}$$
$$STA_{FF}(B_2) = STA_{FFit}(B_2) + DDF_C\big(d(B_2)\big) \tag{13}$$

$$\left.\begin{aligned} STA_{FF}(B_3) &= Greater\ of \begin{Bmatrix} STA_{FF}(B_2) + T_{it} \\ STA_{FFit}(B_3) + DDF_C\big(d(B_3)\big) \end{Bmatrix} \\ &\vdots \\ STA_{FF}(B_N) &= Greater\ of \begin{Bmatrix} STA_{FF}(B_{N-1}) + T_{it} \\ STA_{FFit}(B_N) + DDF_C\big(d(B_N)\big) \end{Bmatrix} \end{aligned}\right\} \tag{14}$$

unsafe operational conditions. Center and TRACON traffic managers work diligently to control this congestion in the TRACON airspace. Analogously, the scheduling algorithm needs a mechanism for controlling congestion of TRACON airspace due to excessive delay buildup.

This step describes an analytical procedure for distributing delay between Center and TRACON airspace. The procedure involves the use of two functions referred to as Center and TRACON delay distribution functions DDF_C *and* DDF_T, respectively, as follows:

$$DDF_C(d) = \begin{Bmatrix} 0 & , & d \le d_{T\max} \\ d - d_{T\max}, & & d > d_{T\max} \end{Bmatrix} \tag{9}$$
$$DDF_T(d) = \begin{Bmatrix} d & , & d \le d_{T\max} \\ d_{T\max} & , & d > d_{T\max} \end{Bmatrix} \tag{10}$$

where $d_{T\max}$ is a parameter that specifies the maximum delay an aircraft is permitted to absorb in the TRACON airspace. As required, the sum of the two functions just equals the delay to be absorbed:

$$DDF_C(d) + DDF_T(d) = d, \tag{11}$$

for all values of d. The two functions are plotted in Figure 2d. These functions are evaluated by substituting into them the delay, d, of each scheduled aircraft as computed by equation (8). Furthermore, the parameter $d_{T\max}$ is itself a function that depends on the meter gate through which an aircraft passes. The meter gate dependency allows modulation of the delay absorption parameter by the length of the nominal (undelayed) path between meter gate and runway. In general, the shorter the nominal path length (more precisely, the TRACON transition time, T_t) the less must be the maximum delay that can be absorbed along that path. In a later chapter, a method for choosing appropriate values for $d_{T\max}$ will be derived.

Step 5: Computing Scheduled Times of Arrival at the Meter Gates

This step describes the procedures for combining the values of the Center delay distribution of step 4, the scheduled times of arrival at the runway of step 3 and the meter gate sequence order of step 2 in order to generate the STA_{FF}'s, the scheduled times of arrival at the meter gates.

In brief, the procedure consists of a push-back of the STA_{FFit}'s, by an amount of time calculated from the Center delay distribution. It may also be thought of as a backward propagation of delays from TRACON to Center airspace. The push-back is done sequentially for aircraft at each meter gate in such a way that the meter gate sequence order is preserved.

The procedure begins with the first aircraft in the landing order. Let that aircraft be B_1, which is consistent with the example sequence in Figure 2e. Then, in accordance

with the definition in equation (6), $B_{11} = C_1$. As the first aircraft scheduled to land, it is always free of delay. The STA_{FF}'s for all the aircraft crossing the meter gate B can then be generated sequentially as shown in equations (12), (13) and (14).

The above series of relations are also used for generating the STA_{FF}'s of aircraft crossing meter gate A. When aircraft are experiencing large delays, the second of the two quantities in the comparison test of equation (14) will be the greater of the two and thus will determine the STA_{FF}. However, in practice, the parameters T_{it} *and* DDF_C can assume such combinations of values that the first quantity becomes the greater of the two. By choosing the first quantity as the STA_{FF} in that case, the logical condition "greater of" ensures that the FCFS meter gate sequence is preserved

The push-back process described here suggests an alternate method for generating a slightly different landing order and scheduled times. Instead of determining the landing order for all schedulable aircraft first, as in step 2, in the alternate method the landing order is generated during the push-back process and is therefore referred to as the push-back adjusted FCFS order method (PAFCFS).

Figure 2e illustrates the graphical construction of the schedule for the PAFCFS method . The push-back of meter gate time is shown in detail for A_1.

In the PAFCFS method, the landing order of the first and second aircraft are generally unchanged.Therefore,the STA'*s and* STA_{FF}'s for these two aircraft are still determined by equations (12) and (13) and their values remain unchanged. To determine the third aircraft to be scheduled to land, select the next aircraft in the FCFS sequence order at each meter gate. Following the example sequence in Figure 2, the next aircraft at gate A is A_1 and at gate B it is B_3. Then compute the in-trail meter gate times for these aircraft:

$$STA_{FFit}\,(A_1) = ETA_{FF}(A_1) \tag{15}$$

$$STA_{FFit}\,(B_3) = STA_{FF}\,(B_2) + T_{it} \tag{16}$$

Next, compute the earliest unconstrained runway times for this pair:

$$RTA(A_1) = STA_{FFit}\,(A_1) + T_t\,(A_1) \tag{17}$$

$$RTA\,(B_3) = STA_{FFit}\,(B_3) + T_t\,(B_3) \tag{18}$$

The next aircraft to be scheduled to land is now chosen to be the one with the earliest RTA, written symbolically as:

Next aircraft to land

$$\{A_1 \text{ or } B_3\} = Arg \text{ (lesser of } \{RTA\,(A_1)\,, RTA\,(B_3)\}) \tag{19}$$

In the example of Figure 2e, the next aircraft is A_1, which represents a change in order compared to the original method. The computation of STA, DDF_C *and* STA_{FF} for the aircraft so selected now parallels the previously described method. Analysis of the PAFCFS order reveals that in comparison to the original order, it tends to advance the landing order of aircraft from gates with lower flow rates relative to those from gates with higher flow rates. While this may be seen as less fair than the original method it also yields on average slightly lower delays.

After these quantities have been computed, they provide the input conditions for equations (15) - (19) to select the next aircraft to be scheduled to land. Thus in contrast to the original method, the landing order here is not known until all aircraft remaining to be scheduled are from a single gate.

EXTENSION TO MULTIPLE RUNWAYS

In this chapter the basic algorithm is extended to handle the scheduling of aircraft to an airport with several landing runways. All large airports use at least two and as

many as four landing runways simultaneously under certain traffic and weather conditions. Similar to the basic algorithm, the objective here is to generate efficient initial schedules for the multiple runway case without time consuming iterative procedures. These schedules and runway assignments can then be used as the starting solution for optimizing procedures if real time computational constraints permit. The guiding principle of the runway assignment process as developed here is to assign and schedule aircraft sequentially to the runway that gives the earliest landing time while minimizing loss of full or fractional landing slots.

First it is necessary to generalize the definition of FCFS at the runway for the multiple runway situation. Begin by completing step 1 for the list of schedulable aircraft. Then compute the unconstrained runway times of arrival, the RTA's, as in equations (3)-(4) of step 2 for each runway. In order to avoid complex symbology, the following development assumes two landing runways, designated as $R1 \; and \; R2$.

Thus, for the gate A aircraft, one can write:

$$RTA_{R1}(A_i) = STA_{FFit}(A_i) + T_{tR1}(A_i) \qquad (20)$$

$$RTA_{R2}(A_i) = STA_{FFit}(A_i) + T_{tR2}(A_i) \qquad (21)$$

and similarly for the gate B aircraft, where the RTA's and $T_t's$ have been appended with the runway identification, $R1(or \; R2)$. Then, for each aircraft, define the preferred runway, RP, as the one having the lesser of the $T_t's$:

$$\begin{aligned} RP(A_i) &= Arg\,\{R1, R2\} \\ &\quad lesser\,of\;\{T_{tR1}(A_i), T_{tR2}(A_i)\} \end{aligned} \qquad (22)$$

and similarly for aircraft from all other gates. Then, the FCFS order is defined as the merged and time ordered set of the corresponding $RTA_{RP}'s$:

FCFS landing order list =

$$\begin{aligned} \{C_p\}_{M+N} &= Arg\{A_{ik}, B_{jl}\} \\ &\quad \left(RTA_{RP}(A_{ik}), RTA_{RP}(B_{jl})\right)_{M+N} \end{aligned} \qquad (23)$$

It would now be possible to generate scheduled landing times by inserting the appropriate minimum time separation, T_{it}, between successive aircraft landing on the same runway, similar to step 3 of the single runway case. While such a schedule is feasible it may also be grossly inefficient for the following reason. At hub airports, traffic arrives in rushes from one or at most two directions, causing the one runway with the shortest transition time between the rush traffic meter gate and that runway, to become overloaded. This would occur because the FCFS order procedure defined above leaves all aircraft assigned to their preferred runways. It is, however, possible to improve upon the preferred runway assignment procedure with little additional computation, thereby providing a more efficient starting condition for subsequent runway assignment optimization steps.

The improved procedure can be used either in conjunction with the preferred runway FCFS order defined above or with the delay distribution adjusted order described in step 5. Assume to start with that the FCFS order of equation (23) is being followed. Let the next aircraft to be assigned and scheduled be A_i from Gate A, and let the next aircraft to cross gate B be B_j. The preferred runways for these two aircraft are $R1 \; and \; R2$, respectively. Then it follows that:

$$RTA_{R1}(A_i) < \;\; RTA_{R2}(B_j) \;\; \text{(FCFS order)} \qquad (24)$$

$$RTA_{R2}(A_i) = RTA_{R1}(A_i) + \Delta_{21}(A_i) \qquad (25)$$

where: $\Delta_{21}(A_i) = T_{tR2}(A_i) - T_{tR1}(A_i) \;\; > \;\; 0$ is the increment in time for A_i to transition from gate A to the non-preferred runway compared to the preferred runway. Corresponding relationships can also be written for B_j. Figure 3 illustrates these concepts for an example sequence.

The next step is to calculate the $STA's$ for all combinations of aircraft next in sequence to cross any of the meter gates and all possible landing runways. Each of the $STA's$ is calculated using the procedure described in step 3:

$$STA_{R1}(A_i) = Greater\ of \begin{Bmatrix} RTA_{R1}(A_i) \\ STA_{R1}(A_{i-1}) + T_{i,i-1}(A_{i-1}, A_i) \end{Bmatrix} \quad (26)$$

The four $STA's$ for the example are shown at appropriate locations on the time line in Figure 3. At any step in the scheduling process, the characteristics of the relationships between values of STA and values of RTA influence the strategy for making efficient runway assignments. Two categories of characteristics can be distinguished each of which exposes particular problems in choosing the aircraft ($A_i\, or B_j$) actually assigned to a runway and scheduled in this step, notwithstanding the assumed preference for FCFS order.

Standard category: Preferred runway $STA's$ are less than non-preferred runway $STA's$ and/or all non-preferred runway $STA's$ are larger than the corresponding $RTA's$.

The runway assignment rule for this category is straightforward. If the FCFS order defined in equation (13) in being followed, then the next aircraft to be scheduled in that order is assigned to the runway giving the earliest STA. If, instead, the pushback adjusted FCFS order is being followed, the RTA's for aircraft yet to be scheduled are updated after an aircraft has been assigned and scheduled. Then the aircraft from the gate yielding the lowest RTA for any eligible runway becomes the next aircraft to be scheduled and is assigned the runway corresponding to the earliest STA. Either scheduling order strategy provides acceptably efficient schedules and runway assignments for this category.

Potential Slot loss category: At least one STA for a non-preferred runway is equal to the corresponding RTA and is less than the preferred runway STA. This case can be written symbolically as:

$$STA_{R2}(A_i) = RTA_{R2}(A_i) \quad (27)$$

$$STA_{R1}(A_i) > RTA_{R2}(A_i) \quad (28)$$

and is illustrated in Figure 3.

The potential slot loss referred to here arises from the fact that scheduling an aircraft to a non-preferred runway incurs an unavoidable delay increment of Δ seconds compared to scheduling it to the preferred runway. Therefore, the quantity Δ establishes the maximum potential slot loss for a non-preferred runway assignment. However, the unavoidable delay increment Δ is not a slot loss if the delay that must be absorbed in assigning an aircraft to a non-preferred runway is larger than Δ for other reasons, such as meeting in-trail separation constraints with a preceding aircraft. The potential for a slot loss exists only when the earliest possible scheduled times to the two runways satisfy the conditions in equation (27). The actual slot loss, $S_l(A_i)$ as distinguished from the maximum potential slot loss is computed as follows:

$$S_l(A_i) = lesser\ of \begin{Bmatrix} \Delta \\ RTA_{R2}(Ai) - STA_{R2}(B_j) - T_{j-1,i} \end{Bmatrix} \quad (29)$$

A value of $S_l > 0$ represents a fractional, or larger, landing slot opportunity that is wasted unless an aircraft from another meter gate is available and can be scheduled instead of A_i to occupy a greater portion of that slot. If such an aircraft is available, for example B_j in Figure 3, then one of the FCFS scheduling order disciplines that had selected A_i as the next aircraft to scheduled would have to be relaxed so B_j can be scheduled instead.

The significance of slot loss derives from its cumulative effect on delays for upstream

aircraft during a period of delay buildup, such as at the beginning of a traffic rush. Under such conditions a slot loss can propagate into additional delays for all aircraft that transfer delays to the next upstream aircraft until a hole occurs in the sequence. Analysis of actual traffic during a rush at a large airport shows that this cumulative effect on delays of the upstream aircraft is between 2 to 4 times as much as the value of the slot loss. Thus, reducing slot loss, especially at the beginning of a rush, gives a large payoff in delay reductions.

The order discipline that would select a candidate aircraft with potentially less slot loss from another gate is one that gives the earliest STA for any eligible aircraft assigned to any runway. The order discipline is referred to as the DDF adjusted STA order. It is the smallest STA in the set generated by application of equation (26) for all aircraft next to cross any gate.

If B_j is the aircraft meeting this order discipline, as shown in Figure 3, then the slot loss for B_j on R2 is:

$$S_l(B_j) = STA_{R2}(B_j) - STA_{R2}(B_{j-1}) - T_{j-1,j} \quad (30)$$

Figure 3 shows it to be zero. Therefore, the conditions for choosing B_j instead of A_i as the next aircraft to be scheduled are:

$$\left.\begin{aligned} STA_{R2}(B_j) &< RTA_{R2}(A_i) \\ S_l(B_j) &< S_l(A_i) \end{aligned}\right\} \quad (31)$$

These conditions are met for B_j as illustrated in Figure 3.

One may ask why the DDF adjusted STA order discipline without the condition in equation (31) should not be used for all aircraft in the schedulable set. The answer is that this order is potentially unstable in that it can produce large and what are considered to be "unfair" position shifts compared to a "fair" FCFS order. It is in fact possible to construct "pathological," but entirely realistic, input ETA_{FF} sequences such that some aircraft from some meter gate will be bypassed (backward position shifted) an indefinite number of times, thereby effectively blocking traffic through that gate, if the DDF adjusted STA order is used exclusively. While the lost slot condition reduces the frequency of excessive backward position shifts, a secure guard against them must be included in the schedule logic by limiting the number of backward position shifts relative to a strict FCFS order to a specified maximum value.. Values of 3-5 would be considered acceptable for the maximum.

The summary, the constructive procedure described above for assigning and scheduling aircraft to runways packs aircraft on runways as tightly as in-trail constraints permit, while also minimizing slot losses. It avoids an unequal buildup of delays between different runways by shifting aircraft to non-preferred runways when it is efficient to do so. It maintains FCFS sequence order at each meter gate and retains that order between each meter gate and runway. It permits a fixed number of positive shifts to occur relative to FCFS order for aircraft from different meter gates if doing so reduces slot losses on the runway.

Simplifying Conditons For Runway Assignment And Landing Sequence Optimization

The problem of landing sequence optimization and, to a lesser extent, runway assignment optimization has been studied by several investigators. Various approaches and solutions are described in the technical literature going back at least 25 years. However, currently known algorithms for generating optimum schedules are computationally slow and therefore are not applicable to real time scheduler design.

Schedule optimization problems are closely related to the well known traveling salesman problem. Both types of problems require combinationally growing search procedures to determine the optimum solution. Such

procedures become computationally impractical to implement in real time applications for all but a small number of schedulable aircraft.

To shed further light on the nature of these problems consider a FCFS ordered set of schedulable aircraft as shown in Table 3. The optimization objective of interest in scheduling is the minimization of the sum of delays of all aircraft by position shifting and runway assignment. No algorithms are known and none are thought to exist that can generate the optimum solution by operating sequentially on this time ordered set, starting with the earliest ETA aircraft. Another interpretation of the non-sequential character of the optimum solution procedure says that the choice of position shifts and runway assignments made at the beginning of the set cannot be made in isolation of, and are therefore interdependent with, such choices at the end of the set.

Now assume that the true optimum solution could be obtained for the whole set of schedulable aircraft converging on an airport by some future superfast computer. Such a solution would, however, be of little practical value in a real time air traffic control environment for two reasons. First unavoidable, unknowable and time varying errors in the computation of the ETA's upon which the optimum solution is based, render the solution non-optimum even if it is computable. Since ETA errors grow approximately proportional with the time-to-fly to the airport, the degree of non-optimality grows with increasing time-to-fly to the meter gate, or airport. Second, even if the optimum solution were known it cannot be enforced because of operational constraints inherent in the human centered air traffic control process. For a variety of reasons, sequencing and runway assignment decisions must be made when an aircraft first reaches a specified time to fly to a meter gate or runway. This time-to-fly parameter is known as the Freeze Horizon. A Freeze Horizon must be established in order to provide controllers with sufficient time and airspace to execute sequencing and runway assignment advisories. Furthermore, controllers require the Freeze Horizon to be held nearly constant, within one or two minutes of a nominal time.

The necessity for a stable Freeze Horizon together with the inevitability of errors in ETA's enable a crucial simplification in the formulation of the scheduling optimization problem. Instead of having to include a large number of aircraft in the combinatorial search as originally thought, thereby creating computational overload, only the few aircraft that, at any time, are within a narrow time range of the Freeze Horizon need to be considered in such a search. In practice, the number of such aircraft can be limited to two or at most three without incurring a significant loss in efficiency.

Thus, in conclusion, a careful examination of the actual operational environment for scheduling and control of arrival traffic permits a simplification of what initially appeared to be an intractable optimization problem to one that is computationally feasible for a real time scheduler.

REAL TIME SCHEDULING ALGORITHM WITH LIMITED SEQUENCE AND RUNWAY ASSIGNMENT OPTIMIZATION

The previous chapter explained the need for incorporating a Freeze Horizon in the design of the real time scheduler. The need for a Freeze Horizon together with unavoidable errors in the ETA_{FF}'s conspired to permit a significant simplification in the runway assignment and sequence optimization problem. This chapter extends the basic algorithm to include both a Freeze Horizon and a limited degree of schedule optimization that is computational tractable in real time.

In addition to the Freeze Horizon, the Optimization Horizon and the Influence Horizon play crucial roles in the real time algorithm. The three horizons segregate the arrival aircraft into four sets based on the values of the ETA_{FF}'s relative to these horizons. The ETA_{FF} time lines in Figure 4 give representative examples of these sets.

Freeze Horizon and Freeze Time-To-Fly
The Freeze Horizon is defined as the sum of current time and a freeze-time-to-fly parameter, which lies in the range of 17-25 minutes. When an aircraft's estimated time-to-fly to the meter gate, as derived from its current ETA_{FF} becomes equal to or less than the freeze-time-to-fly, its runway assignment and landing sequence must be frozen at their last computed values.

Optimization Horizon and Optimization Interval
The difference in time between the Optimization Horizon and the Freeze Horizon equals the Optimization Interval. Runway assignment and sequence optimization will be performed for the first P aircraft with ETA_{FF}'s in this interval. After runway assignments and landing sequences have been determined for these P aircraft, they will be frozen simultaneously. The Optimization Interval has a relatively narrow time range of only 2-5 minutes, reflecting the controller's low tolerance for variability in the location of the Freeze Horizon. The narrowness of the time range also ensures the maximum number of aircraft with ETA_{FF}'s in the Optimization Interval will be small, thus reducing the complexity of the optimization.

Influence Horizon and Influence Interval
The Influence Interval is the difference between the Influence Horizon and the Optimization Horizon. Only aircraft with ETA_{FF}'s less than the Influence Horizon will be allowed to influence the choice of the runway assignments, and landing sequences for the P aircraft in the optimization set. Aircraft with ETA_{FF}'s later than the Influence Horizon are excluded because they are still too far away and, therefore, their ETA_{FF}'s are too uncertain to allow these aircraft to influence the runway assignment process at this time. Their influence will be felt later when these aircraft finally penetrate the Influence Horizon. Experience with the current level of ETA_{FF} accuracy suggests that the Influence Horizon should be located about 10 minutes above (later than) the Freeze Horizon.

The three horizons divide the set of ETA_{FF}'s into four subsets as illustrated in Figure 4. Aircraft below the Freeze Horizon have fixed STA_{FF}'s and runway assignments. In this region controllers handle the aircraft so as to move the ETA_{FF}'s toward coincidence with the corresponding STA_{FF}'s. As aircraft approach the meter gate. Occasionally a controller may invoke commands to unfreeze and then reassign and resequence a particular aircraft or a group of aircraft in the Freeze Interval. Such commands are avoided if possible, because they generally increase workload and create complex control problems.

Three aircraft, $A_{i,} B_{j,} and A_{i+1}$ are located in the Optimization Interval in Figure 4. Runway assignment and sequence optimization is to be performed for the first P of these aircraft. This process is illustrated for P = 2, a realistic value for a real time scheduler. It is carried out in three steps.

The first step generates the set of all runway assignments and scheduling orders for A_i, and B_j , producing what shall be called the comparison set. Since $A_i and B_j$ pass through different meter gates, there are no sequence order constraints to be obeyed at the meter gates and therefore two scheduling orders are possible: $A_{i,} B_j$ and $B_{j,} A_i$.

For each scheduling order all four pairs of runway assignments must be generated. For order A_i followed by B_j they are:

$$A_i \to R1, B_j \to R1$$

$$A_i \to R1, B_j \to R2$$

$$A_i \to R2, B_j \to R2$$

$$A_i \to R2, B_j \to R1$$

These 4 pairs of runway assignments, when combined with the two possible scheduling orders, produce a total of 8 pairs of runway assignments, which constitute the comparison set.

The second step of the process generates the runway STA's for each pair in the comparison set as well as for all other aircraft below the influence horizon. In Figure 4, these other aircraft are A_{i+1}, A_{i+2} and B_{j+1}. If should be noted that they inherited their runway assignments from the initialization procedure previously described, or if none is used, they are assigned to their preferred runways. Figure 4 illustrates one of the eight possible scheduling solutions that are generated in this step. Since runway assignments are fixed for each element in the comparison set, the basic algorithm can be applied to the determination of the STA's in a straight forward way.

The third and final step determines the optimum runway assignment and landing order for A_i and B_j by selecting the minimum delay schedule from among the eight trial schedules of the comparison set. The delay equivalent cost, D, of each trial schedule, k, is defined as the sum of the STA's for all aircraft below the Influence Horizon:

$$\begin{aligned} D(k) &= STA^k(A_i) + STA^k(B_j) \\ &+ STA^k(A_{i+1}) + STA^k(B_{j+1}) \end{aligned} \tag{32}$$

Where in this example, k ranges from 1 to 8. The particular value of the index k that corresponds to the minimum of the D(k) establishes the optimum runway assignment and landing order for A_i and B_j. When this step is completed, the scheduling status of A_i and B_j is changed to frozen. The real time scheduler is now ready to receive a new set of updated ETA_{FF}'s and process them in a similar manner.

Estimating The Number Of Trial Schedules In The Comparison Set

The number of distinct combinations of sequence orders and landing assignments for which trial schedules must be computed was shown in the preceding section to be 8 for the example of 2 landing runways and 2 aircraft in the optimization set. In order to assess the computational load for other cases of interest, it is useful to estimate the number of such trial schedules in the comparison set. If no limit is placed on the number position shifts allowed, then the number of scheduling orders is P! for P for aircraft in the optimization set. It should be noted that the scheduling order is the same as the landing order.

Let Q be the number of landing runways. Since each aircraft in a scheduling order of P aircraft may be independently assigned to any of the Q runways, the number of possible runway assignments for each scheduling order is Q^P. Therefore an estimate of what is essentially an upper limit of the number of trial schedules K, that the scheduler must compute to locate the optimum is:

$$K = P!\,Q^P \tag{33}$$

Clearly, K exhibits an extremely fast growth rate even for small increments in P and Q. For example if P and Q are both increased from 2 to 3, K increases from 8 to 162, which is too large to be handled by a real time scheduler.

Limiting the position shifts to 2 reduces k to 81 for this example, but even this number of trial schedules is too large to be evaluated in real time. A current software implementation of the basic algorithm, which handles assignments to three runways, is designed for the P=1 case, and thus needs to generate only three trial schedules.

A modest improvement in scheduling efficiency can be obtained, especially for the P=1 case, by following the runway assignment of the freeze aircraft with a single position shift trial involving the freeze aircraft and the last-to-freeze aircraft. However the delay reduction potential of position shifting is somewhat reduced when it follows runway assignment optimization. This occurs because runway assignment optimization tends to assign aircraft with similar weight classes to the same runway, thus obviating the advantage of position

shifting for some situations. Nevertheless it still yields worthwhile benefits.

Adding A New Aircraft To The Schedule

The addition to the basic algorithm that optimizes the schedules of aircraft near the freeze horizon and then transitions them from non-frozen to frozen status, the real time scheduler also contains numerous functions for handling a variety of special scheduling events. Such events can be triggered by commands from controllers or by inputs from other components of the automation system. For example, a controller may issue a command to reschedule an already frozen aircraft or reassign a group of frozen aircraft to a different runway. To handle the more complex events, for example runway configuration changes, the basic algorithm must be modified significantly. The management of these events in real time and the synthesis of algorithms to generate the proper responses increase the complexity of the final software design by an order of magnitude, (measured by lines of computer code) compared to the software design of the basic scheduling algorithm alone. Thus the software implementation of the full function scheduler based on the algorithm described in this paper contains about 45,000 lines of C code.

This section describes a modification to the basic algorithm for handling one of the most important as well frequently occurring special events; the arrival of a new aircraft. This event is signaled to the scheduler by the aircraft tracking and trajectory analysis modules of the automation system. The essential data associated with the events are comprised of the aircraft identifier, the arrival meter gate and the ETA_{FF} for the newly born aircraft. The scheduler must respond by adding this aircraft to the list of scheduled aircraft in a fair and efficient manner.

The procedure for adding the newly born aircraft is a variation of the basic algorithm. First, the aircraft is merged with the existing set of active aircraft in FCFS meter gate sequence order. This is illustrated in Figure 5 for A_{new}. Second, it is scheduled to the meter gate behind its lead aircraft, A_{i+1} in Figure 5 using the applicable meter gate in-trail constraint T_{it}. Third, starting at the meter gate time $STA_{FFit}(A_{new})$ aircraft A_{new} is scheduled to each of the available landing runways at the earliest time that is consistent with applicable meter gate-to-runway sequence constraints <u>and</u>, in addition, is behind the last frozen aircraft. This creates the two trial STA's shown in Figure 5. Thus, on R1, A_{new} has to follow A_{i+1} with the appropriate time separation. On R2 it would have to follow B_j since the status of B_j was changed to frozen after the assignment process described in the preceding section was completed. Fourth, for each of the two trial STA's, the corresponding $STA_{FFR1}(A_{new})$ and $STA_{FFR2}(A_{new})$ are determined by applying the required delay distribution. Fifth, all old aircraft behind (A_{new}) in meter gate sequence order and below the influence horizon are rescheduled to their previously assigned runways. The rescheduling must include the appropriate delay feedback. Sixth, the runway assignment for A_{new} is now determined by evaluating the delay-equivalent cost function, equation (31), for the two trial assignments, and choosing the assignment giving the lowest cost.

In summary, during the flight history of an aircraft in Center Airspace beginning with the start of active tracking and ending at the time of meter gate crossing, the scheduler makes runway assignments for each aircraft twice. The first time is a preliminary assignment done at the start of active tracking. It ensures that every aircraft in the current schedulable list has an appropriately assigned runway. This permits the scheduler to generate what might be called pseudo schedules, so named because they are never actually controlled to, but are used only to provide continuously updated estimates of expected delays. Controllers use these estimates, displayed in graphically convenient form, to formulate control strategies. The second time the assignment

is made takes place just before the freeze and involves the optimization procedure described previously. However, it should be noted that the first assignment also influences the outcome of the optimization procedure because aircraft below the influence horizon retaining their original assignment still contribute to the value of the cost function given in equation (32).

While runway assignments are computed only twice, the STA_{FF}'s are updated every 10-15 seconds prior to freeze. Experience with operating this scheduling algorithm in live traffic has shown that this update strategy achieves an appropriate balance between stability and responsiveness of the schedule to ETA_{FF} updates.

When an aircraft crosses a meter gate and enters TRACON airspace, it comes under the control of TRACON automation-tools, such as the Final Approach Spacing Tool (FAST). At this time the aircraft is unfrozen and the FAST scheduler makes the final runway assignment. If the traffic is being controlled accurately to the meter gates, the final assignment will, more often than not, be the same as the previous assignment. The next chapter will examine the impact of control accuracy on the design of the scheduler in detail.

STRATEGY FOR DELAY ABSORPTION IN THE PRESENCE OF TIME CONTROL ERRORS

Whenever arrival traffic demand exceeds aircraft landing capacity over a 15 minute or longer time interval a significant buildup of delay is likely to occur. After years of experience in dealing with such situations at large airports, traffic managers have learned how to anticipate the magnitude of a delay buildup and have devised standard procedures for absorbing the delay.

While traffic management procedures in use today generally achieve smooth traffic flow even when delays have built up significantly, controversy lingers over what is the best procedure for delay absorption. The dividing chasm in the controversy is between pilots and airline operators on the one hand and controllers and traffic manager on the other.

Pilots and airline operators prefer delays to be absorbed close to the airport even to the point where holding is required in the TRACON airspace at low altitude. They fear that early delay absorption far from the airport does not produce sufficient traffic pressure to achieve a high landing rate.

Traffic managers and controllers, on the other land, contend that, on balance, it is more efficient to absorb most delay in the Center airspace far from the airport so as to maintain traffic flow in the TRACON smooth and orderly. They further contend that delay absorption strategies that lead to frequent holding in the TRACON airspace create high workload for controllers and risk chaotic traffic conditions that actually reduce landing rates.

The structure of the basic scheduling algorithm described in the preceding chapters, when analyzed in combination with models of aircraft fuel consumption and accuracy of time-control at the Center-TRACON boundary can provide a rational solution to the delay absorption controversy. The solution derives from a method of analysis that determines the value of delay distribution between Center and TRACON airspace such that the average direct operating cost of delay absorption for the arrival traffic is minimized.

As indicated above, the two factors that are the key to the analysis are aircraft fuel consumption and accuracy of time control. It is well known that the minimum fuel flow rate (lbs/sec) of turbofan powered aircraft is significantly less at cruise altitude than it is at sea level altitude. Therefore, it is more fuel efficient to absorb delays at or near cruise altitude than it is at sea level. The performance manual of an aircraft contains the basic data needed to derive the relationship between fuel consumption and delay absorption at high and low altitude.

Such a relationship has been derived below for a Boeing 727 aircraft:

$$F = \left(120 d_c + 180 d_T\right)\frac{1}{60} \qquad (34)$$

where d_c is the Center delay, which is assumed to be absorbed at 30,000 ft and d_T is the TRACON delay assumed to be absorbed at 3,000 ft. The quantity F is the additional fuel consumed in lbs. due to delays d_c and d_T given in units of seconds. If the total delay to be absorbed is $d = d_c + d_T$, then equation (34) shows that choosing $d = d_c$ and $d_T = 0$ minimizes the additional fuel consumption for any delay d. It therefore follows that if the total delay to be absorbed can be determined when the aircraft is still at or near cruise altitude and a method for controlling the delay exists, the most fuel efficient and, therefore, cost efficient strategy is to absorb all delay in the high altitude Center airspace and none in the low altitude TRACON airspace.

This result leads directly to the question of how the inevitable limitations in the accuracy with which delays can be absorbed in Center airspace should change the proposed delay absorption strategy.

This question is illuminated by examining the operation of the real time-scheduler. In the scheduling algorithm described in the preceding chapter, the final value of required delay absorption is determined at the time an aircraft's STA_{FF} is frozen. This occurs at the freeze horizon when an aircraft is approximately 19 minutes of flying time from its assigned meter gate. The meter gate, located at the boundary between Center and TRACON airspace, therefore provides the dividing point for distributing the total delay between Center and TRACON airspace. And the delay distribution function, DDF, which is imbedded in the architecture of the basic real time algorithm, provides the mechanism for allocating the delay to each airspace on an aircraft by aircraft basis.

Thus the basic information needed to study the question posed above is to determine the expected accuracy of controlling an aircraft to cross the meter gate at time STA_{FF} assuming the aircraft is initially 19 minutes away from the meter gate when the control process begins.

Accuracy of control for both 19 and 30 minutes of flying time to the meter gates was recently estimated by analyzing over 3000 actual flights that landed at the Dallas/Fort Worth Airport. The estimates of accuracy were determined both for the metering system currently in use at the Fort Worth Center, and for the Center/TRACON Automation System which is scheduled for field tests at the Fort Worth Center. A NASA report by Mark Ballin and the author describing the accuracy analysis is in preparation [11]. Of particular interest here are the two standard deviation errors at 19 minutes to the meter gates for the current metering system and for CTAS. They are, respectively, 180 and 90 seconds. Before developing a numerical technique for studying the effects of these errors, it is important to understand qualitatively why these errors will have an adverse effect on system performance. The adverse effect can be summarized succinctly as slot loss. It is most easily visualized when traffic is dense and all delay is being absorbed in the Center airspace. If an aircraft crosses the meter gate later that its prescribed STA_{FF}, all aircraft scheduled behind the late aircraft at the minimum time separation will have this time error also passed on to them, similar to how a falling domino topples the next one. Since the time to fly from the meter gate to the runway is, by assumption equal to the minimum time, it is impossible to recover this slot loss completely by speeding up or short cutting the path. Moreover, similar to the runway assignment problem previously described, the total delay increment due to a fractional slot loss can be a several times the magnitude of the original time error if the error is propagated to several trailing aircraft. Thus, the putative benefits in fuel efficiency of absorbing all delays in Center airspace are being eroded by delay increments and the resulting fuel losses due to those meter gate crossing time errors. While it is true that perceptive pilots and controllers have anecdotally referred to this

phenomenon, quantitative studies on it have not been done to the author's knowledge.

Stochastic Simulation Of Meter Gate Crossing Errors

The effect of meter gate crossing errors was studied quantitatively by stochastic Monte Carlo simulation developed by Frank Neuman et al. and described in several NASA reports [9]. A simplified diagrammatic representation of this simulation is shown in Figure 6. The upper part of the figure represents the basic scheduling algorithm. The diagram draws attention to the two distinguishing characteristics of the algorithm, namely the delay distribution function for allocating delays and the feedback-like effect of this function through the sequential pushback of the STA_{FF} 's. The input to the algorithm is a set of ETA_{FF} 's representing the simulated traffic scenario. They are generated by a random process that has been carefully designed to match the statistical characteristics of a typical 90 minute long traffic rush at the Dallas/Fort Worth airport. The simulation drives the algorithm with several thousand samples of such traffic rushes, all different from each other, yet statistically identical. The performance of the algorithm is measured by calculating delay and fuel consumption averages for thousands of such rush traffic samples. Although the input traffic is statistically generated , this part of the simulation produces a deterministic set of STA_{FF}'s and STA's for each randomly generated set of ETA_{FF}'s.

The lower part of Figure 6 represents the stochastic simulation of meter gate crossing time errors. The simulation generates an actual time of arrival, ATA_{FF}, over a meter gate for each aircraft by adding a randomly generated meter gate crossing time error, N_{pc} to each aircraft's STA_{FF}, the latter being provided by the simulation of the basic scheduling algorithm. The statistical properties of N_{pc} are chosen to match the empirically determined probability distribution of meter gate crossing errors. Although the errors were found to be nearly normally distributed, they are approximated here by the convolution sum of three uniformly distributed random variables having the general shape shown in the figure. This approximation eliminates the somewhat unrealistic, for this problem at least, tail values found in the normal distribution. The ATA_{FF}'s now provide the input to what is referred to in the figure as the TRACON scheduler. This scheduler is identical to the basic scheduler but with $d_{T\max}$ set to zero. By reassigning and resequencing aircraft at the time they actually cross the meter gates, the TRACON scheduler compensates, to the degree that is possible, for the adverse effects of the meter gate crossing errors. Moreover, the twice repeated application of the sequencing and runway assignment algorithm, first at the Center freeze horizon and than at the TRACON boundary, represents the actual operation of CTAS as it is being implemented in the field.

The output of the two parts of the simulation, where the output of the first becomes the input to the second, generates runway threshold STA's whose values accurately reflect both the efficiencies gained by sequencing and runway assignment optimization as well as the penalties imposed by the pilot- controller errors in meter gate crossing times.

Analysis of Results

The stochastic Monte Carlo simulation tool briefly described in the preceding section will now be used to investigate the quantitative relationship between delay distribution strategies, meter gate time control errors and scheduling efficiency.

These relationships will be presented here for the single runway case. This case is not only important in its own right, but it also reveals the essential characteristics of these relationships more clearly than the multi-runway case. The multi-runway case, though qualitatively similar, is somewhat more complex to explain and will be covered in a NASA report.

The route structure modeled in the simulation consists of four meter gates with two independent traffic streams converging on each gate. One stream contains a mix of large and heavy jets, the other only large turboprops. The streams are independent by virtue of a required large altitude separation between them at the crossing point. Independence implies that there are no in-trail separation restrictions between aircraft in different streams converging on the same gate. The input traffic rate is 36 aircraft/hour, which is slightly above the maximum sustainable traffic level. There will thus be a significant build up of delays at this traffic level. All data points used in plotting of curves represent averages over 1000 randomly generated traffic samples, each of which contains 54 aircraft in a 90 minute rush period, or 36 aircraft/hour.

The results, plotted in Figures 7-9, focus exclusively on the effects of meter gate crossing errors. The first of the figures, Figure 7, plot the delay increment Δd as a function of the TRACON delay distribution variable, $d_{T\max}$, with meter gate crossing errors, N_{pc}, as a parameter. It is seen that the origin of coordinates corresponds to $\Delta d = 0, d_{T\max} = 0$ *and* $N_{pc} = 0$. The average delay obtained for the simulated traffic scenario at these ideal operating conditions was found to be 280 seconds.

For each of the three non-zero values of N_{pc} the delay increment Δd, decreased strongly with increasing values of $d_{T\max}$. For the highest value of N_{pc}, 180 seconds, which corresponds to the crossing errors of the current operational systems, the reduction in the delay increment is especially striking, declining from 80 seconds at $d_{T\max} = 0$ to only 11 seconds at $d_{T\max} =$ 180 seconds. This result clearly confirms the ability of TRACON delay distribution to compensate almost completely for slot loss due to meter gate time control errors. At the two lower values of N_{pc}, the delay increments are less to start with and decline to correspondingly lower values as $d_{T\max}$ is increased. The N_{pc} = 30 seconds case establishes the practical lower limit of errors, which would be reached when the CTAS Descent Advisor (DA) becomes operational in Center airspace. The middle value of N_{pc} = 90 seconds can be achieved with the CTAS Traffic Management Advisor. At $N_{pc} = 0$, delay distribution has no effect on delay increment, as expected.

The asymptotic limits of this family of curves suggest a simple rule of thumb for choosing the optimum delay distribution. It is to choose $d_{T\max}$ equal to N_{pc}. There is, however, a practical upper limit on $d_{T\max}$ of about 100 seconds that prevents the selection of the optimum value for N_{pc}> 100 seconds. The upper limit reflects the limitations on the availability of airspace within the TRACON to perform complex delay maneuvers.

A significant difference in the effect of TRACON delay distribution exists between the single and multi-runway cases. In the multi-runway case, a non-zero $d_{T\max}$ helps to reduce delays even for $N_{pc} = 0$. Analysis of this case shows that delay distribution in the TRACON mitigates the effects of meter gate in-trail constraints and potential for slot losses and, therefore, delays, when aircraft are assigned to non-preferred runways. This case will be examined in a future NASA report.

Finally, this result does support the opinion of those that believe allocating large delays to the TRACON minimizes slot losses.

A substantially different picture emerges from Figure 8, which plots the increment in fuel consumption ΔF as a function of the same two variables as in Figure 7. The incremental fuel consumption at $d_{T\max} = 0$ and N_{pc}=180 seconds is remarkable for its magnitude, which is 230 pounds for the average aircraft in the traffic sample. This represents a significant economic penalty in

fuel consumption resulting directly from time errors at the meter gates. Initially the fuel consumption strongly declines as $d_{T\max}$ increases. However, the distinguishing feature of the curves is that they reach a clearly defined minimum with respect to the variable $d_{T\max}$. Beyond the minimizing value of $d_{T\max}$ the fuel consumption begins to rise again and becomes asymptotic to the $N_{pc} = 0$ curve. In this case, the rule of thumb for choosing the fuel optimum value of $d_{T\max}$ is $d_{T\max}$ = 2/3 N_{pc}. This result reflects the influence of the fuel consumption trade off relation, equation (34). It shows that high values of TRACON delay distribution exact a fuel cost penalty that weighs against the benefits of incremental delay reduction shown in Figure 7.

This result gives support to the opinion of those who believe that a large amount of delay allocation in the TRACON can have adverse effects. However, the explanation for these adverse effects given here differs in essential ways from the anecdotal arguments that have heretofore been advanced against large TRACON delay distributions.

Introduction To A Unification Principle Of Delay Distribution

The conundrum of delay distribution exposed in the preceding section has a rational resolution originating in the definition of direct operating cost, a widely used measure in the economics of airline operations. Direct operating cost, DOC, is commonly defined as the sum of the cost of time and the cost of fuel as follows:

$$DOC = TC_T + FC_F \tag{35}$$

where T is the time to fly a trajectory in seconds, F is the fuel consumption of a trajectory in lbs. and C_T and C_F are cost factors for converting time and fuel to DOC measured in dollars. Airline operations analysts can provide data for deriving the values of cost factors $C_T \, and C_F$, applicable to the average aircraft in an airline's fleet. Such data were obtained from a large US airline whose aircraft fleet can be approximated by a Boeing 727. From this data the following relationship was derived:

$$F = 10DOC - 2T \tag{36}$$

The choice of F as the dependent variable anticipates the use of equation (36) in the analysis to follow.

To prepare for the application of equation (36), Figures 7 and 8 have been combined in a two parameter family of curves sometimes referred to as a carpet plot. In this carpet plot, Figure 9, fuel and time increments may both be considered dependent variables plotted along vertical and horizontal axes, respectively. The independent variables are the parameters $d_{T\max}$ and N_{pc}.

The unification principle may now be defined as the process by which the carpet plot of fuel and time increments is combined with the time-fuel-DOC relationship given by equation (36) to select the delay distribution strategy that minimizes the increment in direct operating cost for the average aircraft during the rush traffic period. Note that since equation (36) is linear in all variables, incremental variables can directly replace the original variables in equation (36) without changing its form.

The process can be understood by super imposing the DOC increment curves derived from equation (36) on the time-fuel coordinates of Figure 9. Then it can be shown that the unification principle is satisfied at the point of tangency of a linear DOC curve with a specific N_{pc} curve. The value of DOC that produces tangency to the curve of a selected value of N_{pc} gives the lowest possible DOC increment corresponding to that value of N_{pc}. It therefore defines the optimum operating point for the selected value of N_{pc}. The final step is to select the delay distribution parameter, $d_{T\max}$, corresponding to the optimum operating point. That is done by

interpolating on values of constant $d_{T\max}$ curves to find a curve that passes through the optimum operating point. That value of $d_{T\max}$ establishes the optimum delay distribution, d_{Topt}. For the case of $N_{pc} = 180$ sec, $d_{Topt} = 120$ sec and for $N_{cp} = 90$ sec,, $d_{Topt} = 60$ sec.

The difference in the incremental *DOC* for any two values of meter gate time-errors has an important interpretation. It represents the cost penalty of operating an air traffic control system at the higher meter gate time-error compared to operating it at the lower value. Conversely, this difference also give the average cost saving per landing that would be obtained by implementing a new technology that reduces the meter gate time-errors by a specified amount. For example, by reducing the time error from the current value of 180 seconds to 30 seconds attainable with the DA tool the cost savings for each landing aircraft would average 14 dollars.

Finally, the analysis in this section has resolved the long stranding conundrum of how to choose the optimum delay absorption strategy.

REFERENCES

[1] Erzberger, Heinz, and Tobias, Leonard: A Time Based Concept for Terminal-Area Traffic Management, in "Efficient Conduct of Individual Flights and Air Traffic," pp 52-1 - 52-14, AGARD CD-410, June 1986.

[2] Slattery, R.A., "Terminal Area Trajectory Synthesis for Air Traffic Control Automation," Conference Proceedings of the American Control Conference, June, 1995.

[3] Erzberger, Heinz and Nedell, William: Design of Automated System for Management of Arrival Traffic, NASA TM 102201; June 1989; Engle, Laurie: Conflict Detection Tool, Addendum to TM 102201, Oct. 1989, NASA Ames Research Center.

[4] Erzberger, H., Davis, T.J. and Green, S.M., "Design of Center-TRACON Automation System," Proceedings of the AGARD Guidance and Control Panel 56th Symposium on Machine Intelligence in Air Traffic Management, Berlin, Germany, 1993, pp. 52-1 - 52-14.

[5] Davis, T.J., Erzberger, H., Green, S.M., and Nedell, W., "Design and Evaluation of an Air Traffic Control Final Approach Spacing Tool," Journal of Guidance, Control, and Dynamics, Vol. 14, No. 4, July-August, 1991, pp. 848-854.

[6] Green, S.M., Vivona, R.A., Sanford, B., "Descent Advisor Preliminary Field Test," Proceeding of the AIAA Guidance, Navigation, and Control Conference, Baltimore, Maryland, August 7-9, 1995.

[7] Davis, T.J., Krzeczowski, K.J. and Bergh, C.C., "The Final Approach Spacing Tool," Proceedings of the 13th IFAC Symposium on Automatic Control in Aerospace, Palo Alto, California, Sept. 1994.

[8] Krzeczowski, K.J., Davis, T.J., Erzberger, H., Lev-Ram, I. and Bergh, C.C., "Knowledge-Based Scheduling of Arrival Aircraft in the Terminal Area," Proceedings of the AIAA Guidance, Navigation, and Control Conferences, Baltimore, Maryland, August, 1995.

[9] Neuman, Frank and Erzberger, Heinz: Analysis of Delay Reducing and Fuel Saving Sequencing and Spacing Algorithms for Arrival Traffic, NASA TM 103880, Oct. 1991, NASA Ames Research Center.

[10] Brinton, C.R., "An Implicit Enumeration Algorithm for Arrival Aircraft Scheduling," Proceedings of the 11th Digital Avionics Systems Conference, Seattle, Washington, Oct., 1992.

[11] Ballin, M.G., Erzberger, H., "An Analysis of Aircraft Landing Rates and Separations at Dallas/Ft. Worth Airport," to be published as a NASA TM, Fall, 1995.

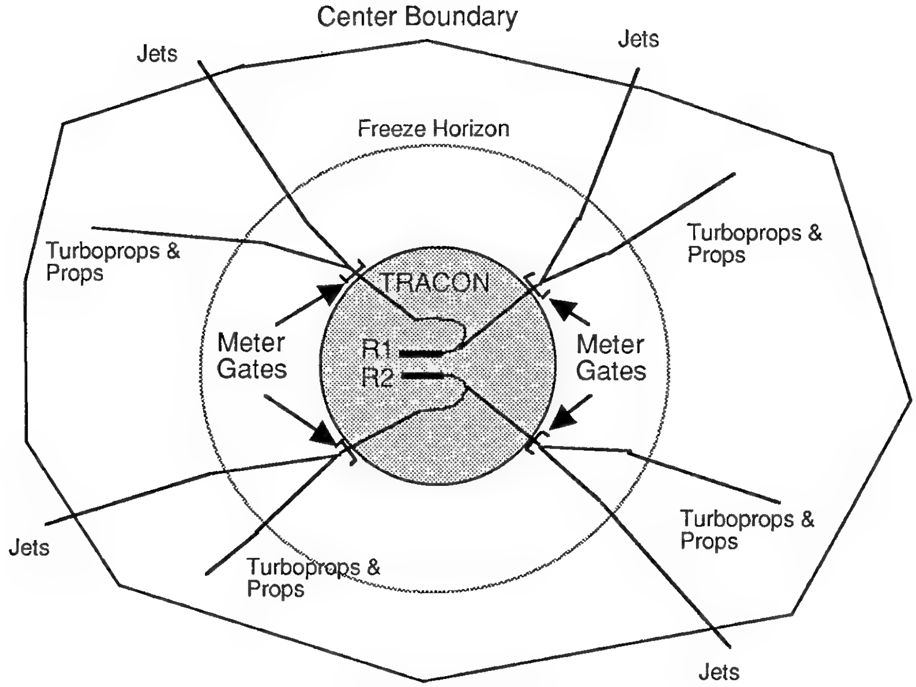


Figure 1 - Airspace Structure and Arrival Routes

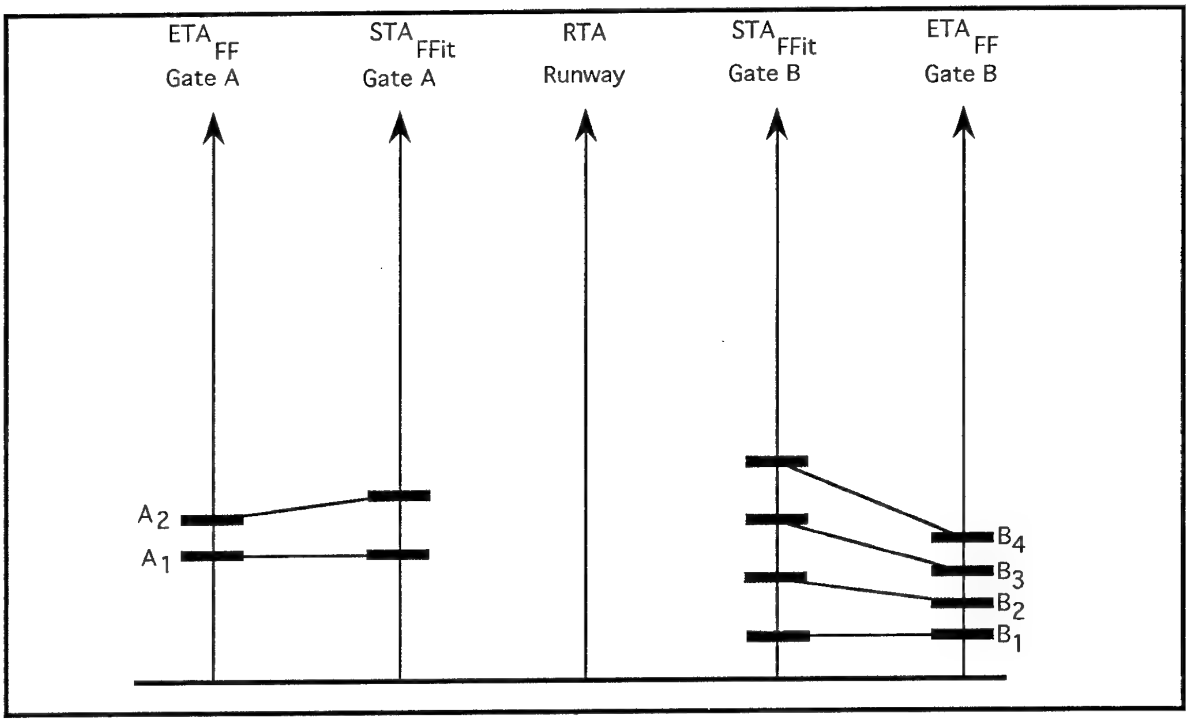


Figure 2. Basic Scheduling Algorithm:
(a) Adding in-trail constraints at the meter gates

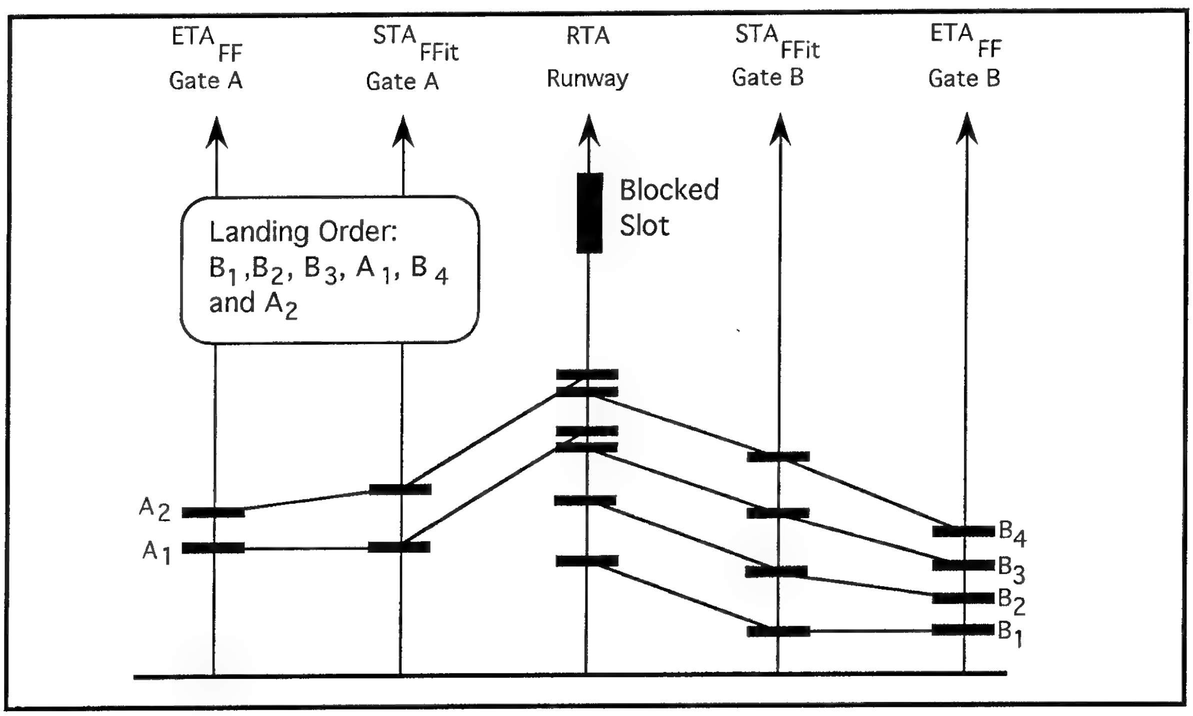


Figure 2 b. Determining Landing Order

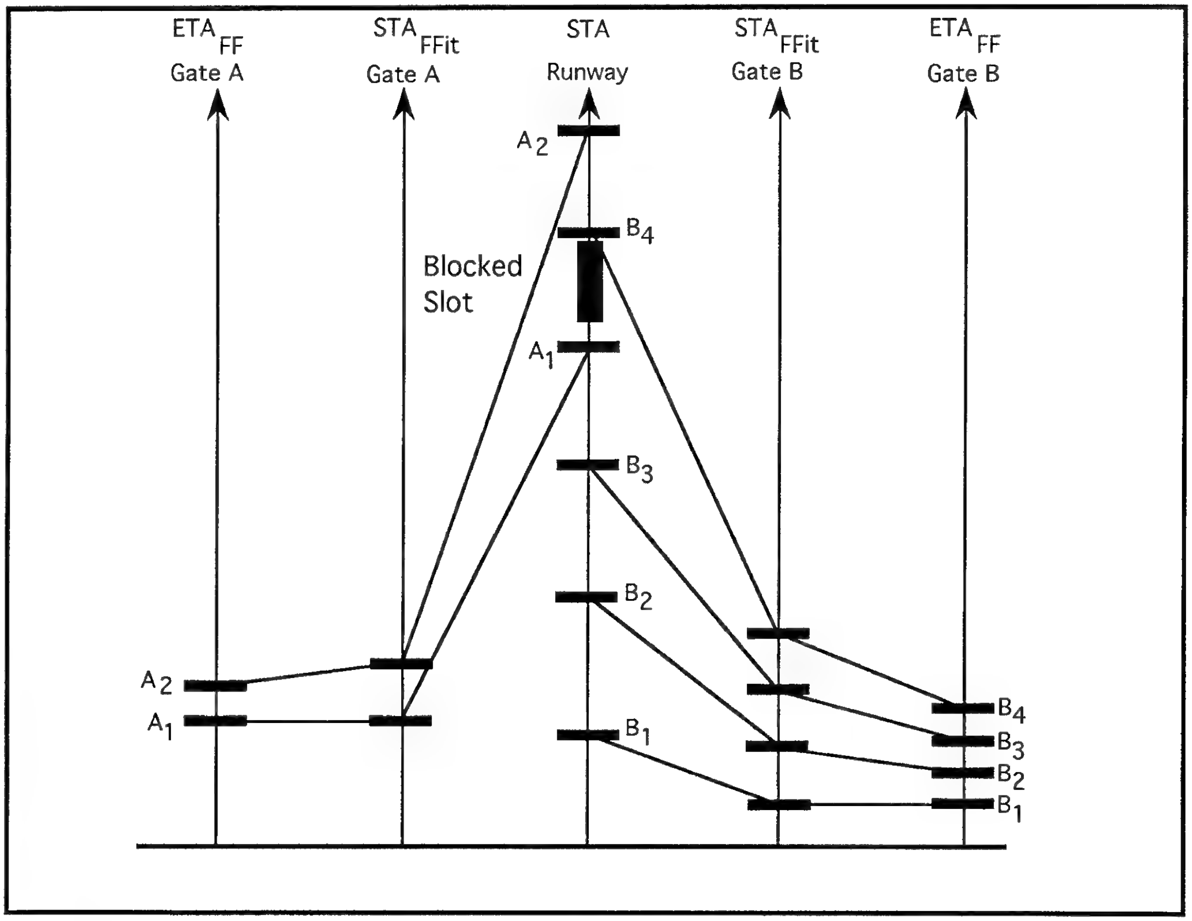


Figure 2 c. Determining the Runway STAs

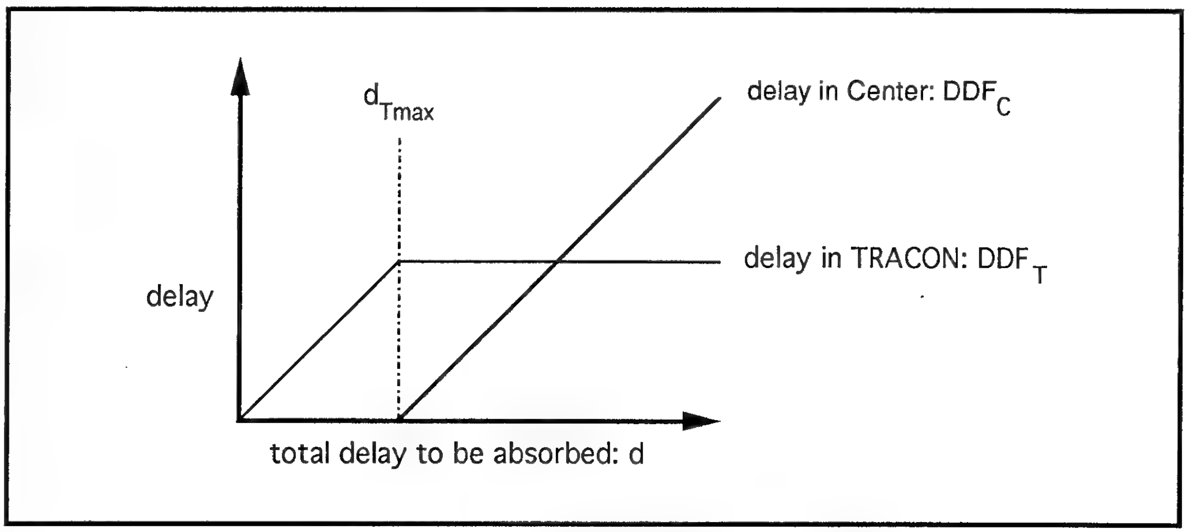


Figure 2d - Delay distribution function

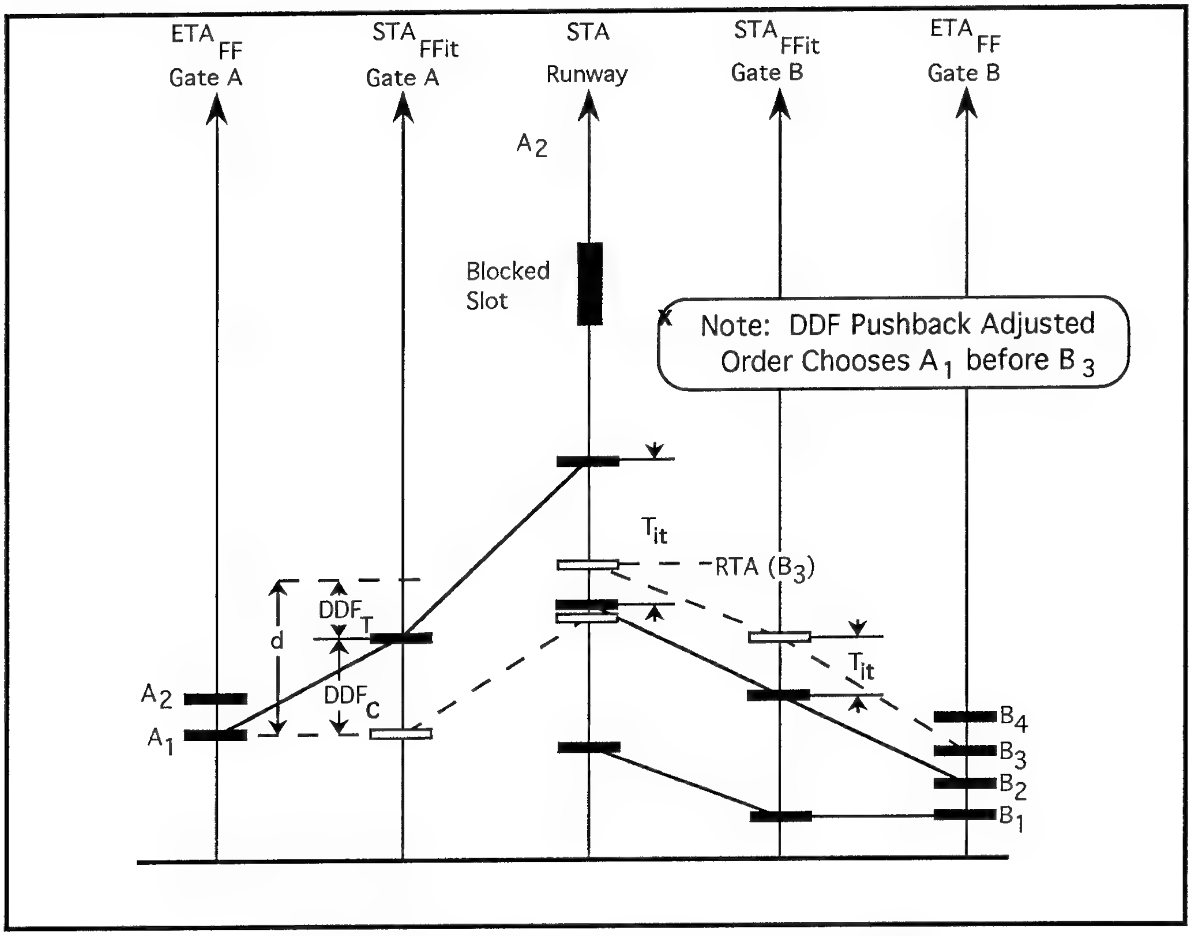


Figure 2 e. Pushback of STA $_{FF}$'s using DDF

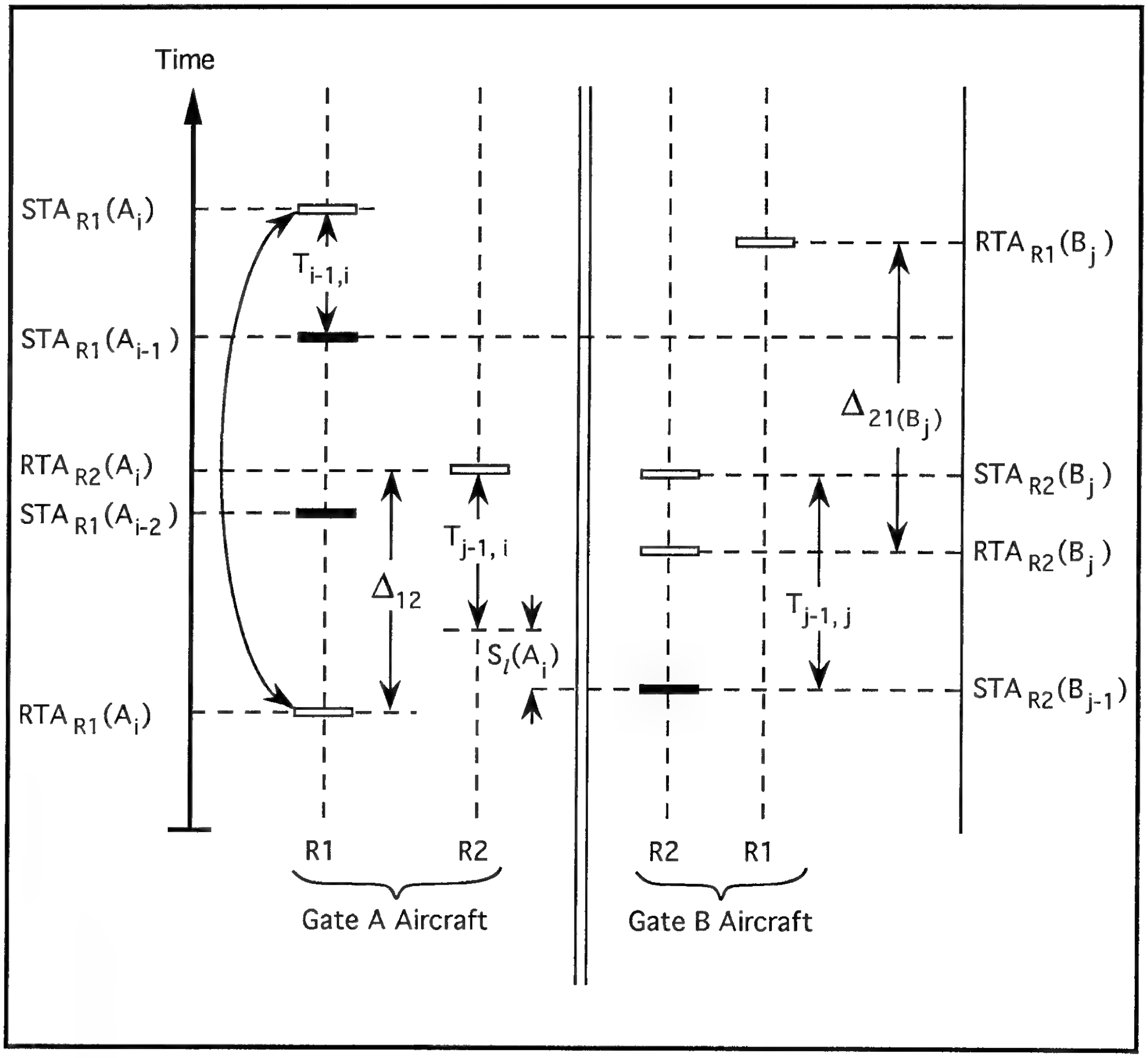
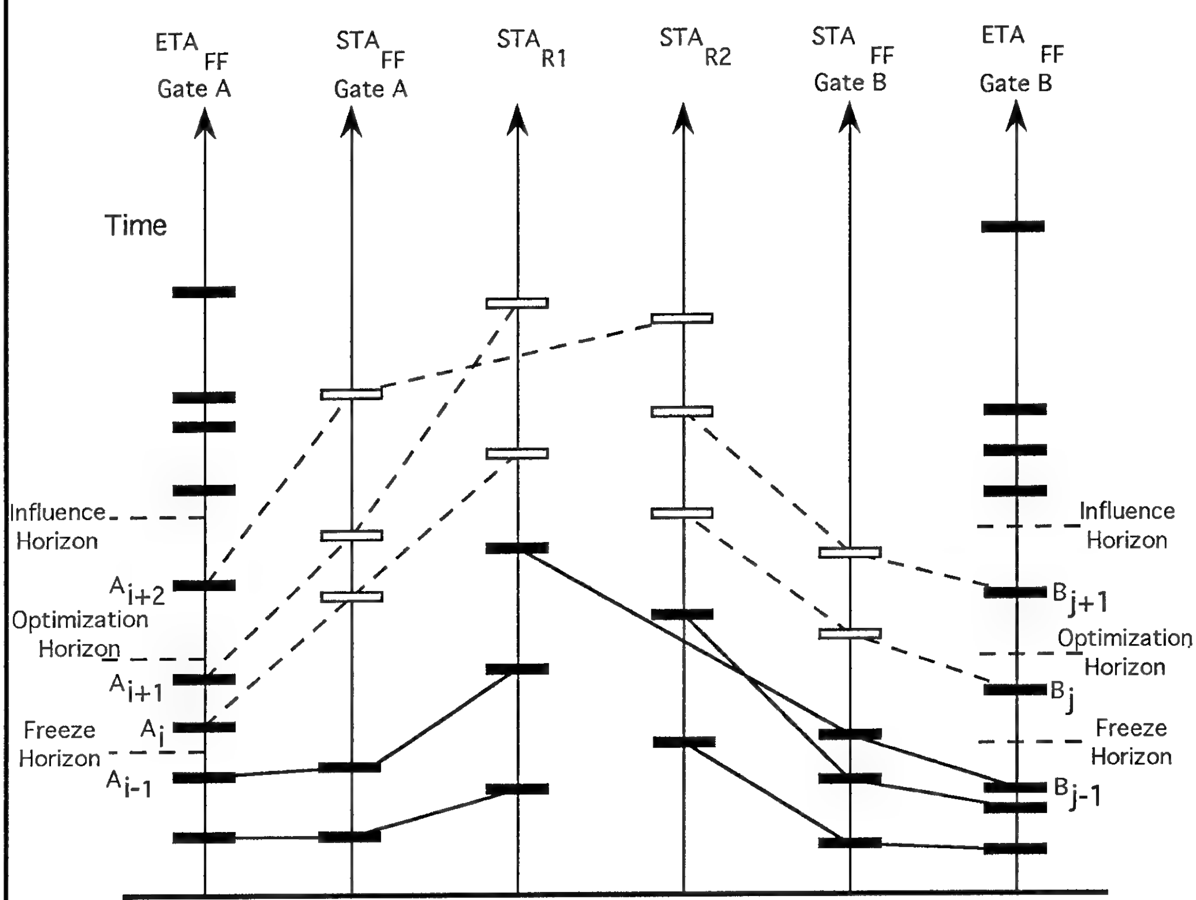


Figure 3 . Choosing a non-FCFS order and runway assignment that minimizes slot loss



Unfilled boxes and dashed lines indicate one of the eight possible trial schedules generated in finding the optimum runway assignments for A_i and B_j .

Figure 4 . Illustration of real time scheduling algorithm

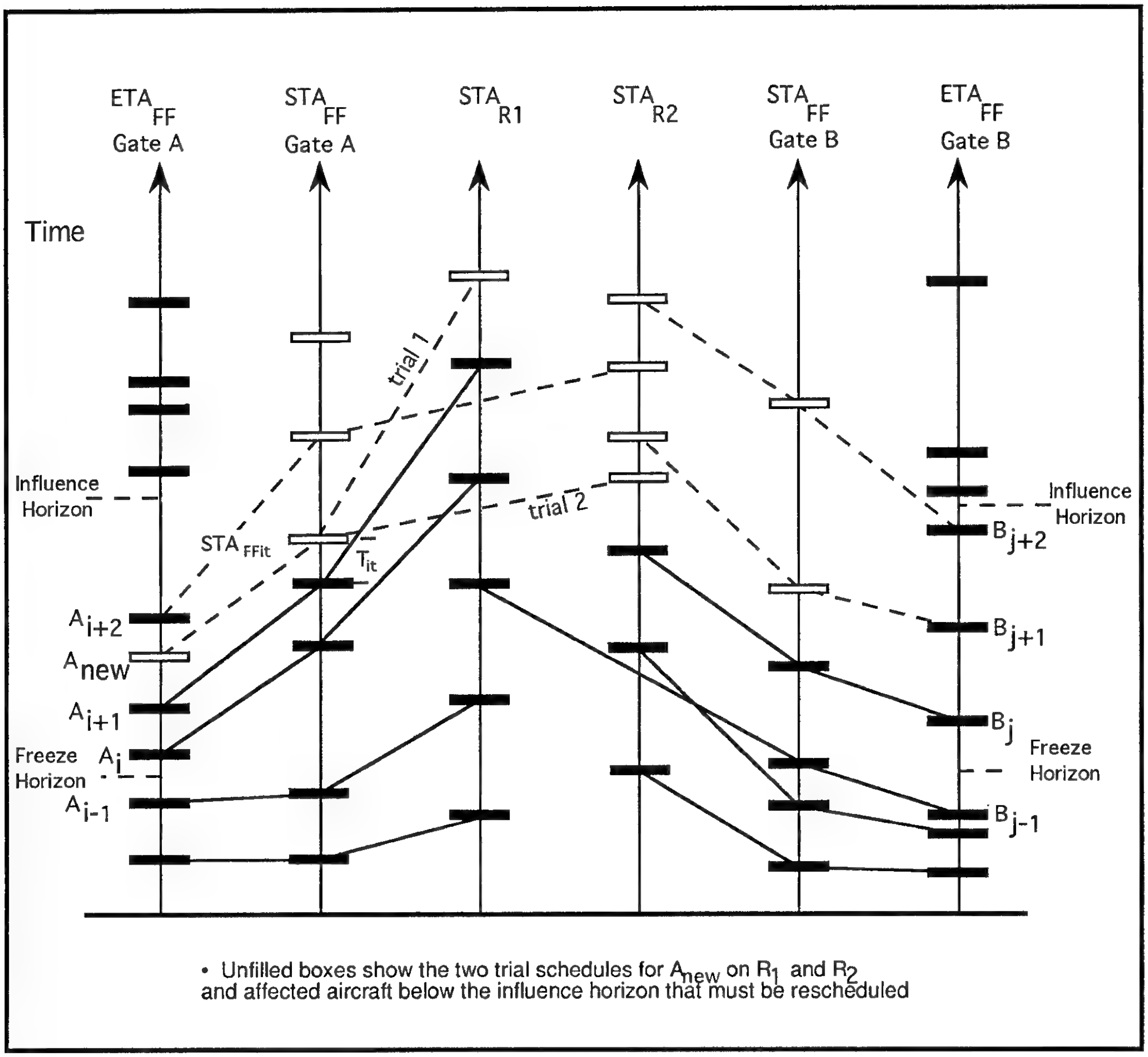


Figure 5. Illustration of real time scheduling algorithm: Adding a new aircraft to the scheduled list

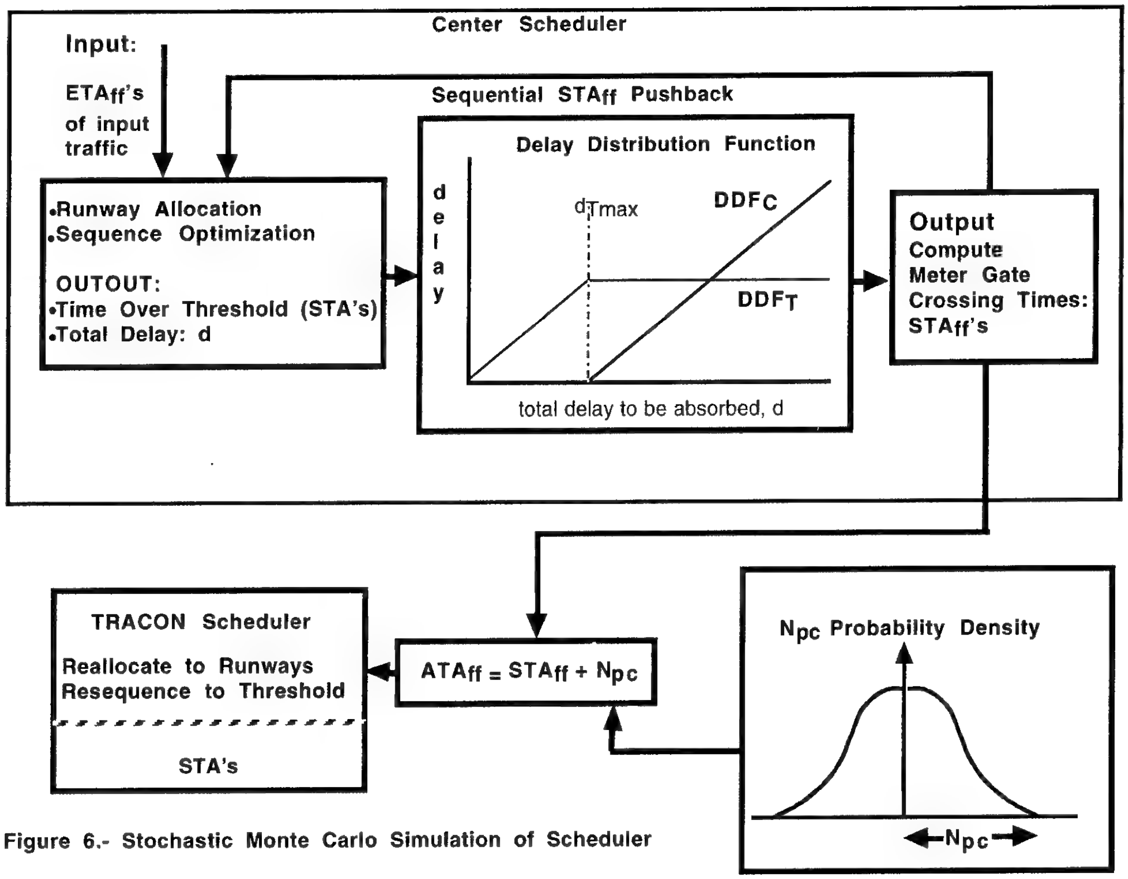


Figure 6.- Stochastic Monte Carlo Simulation of Scheduler

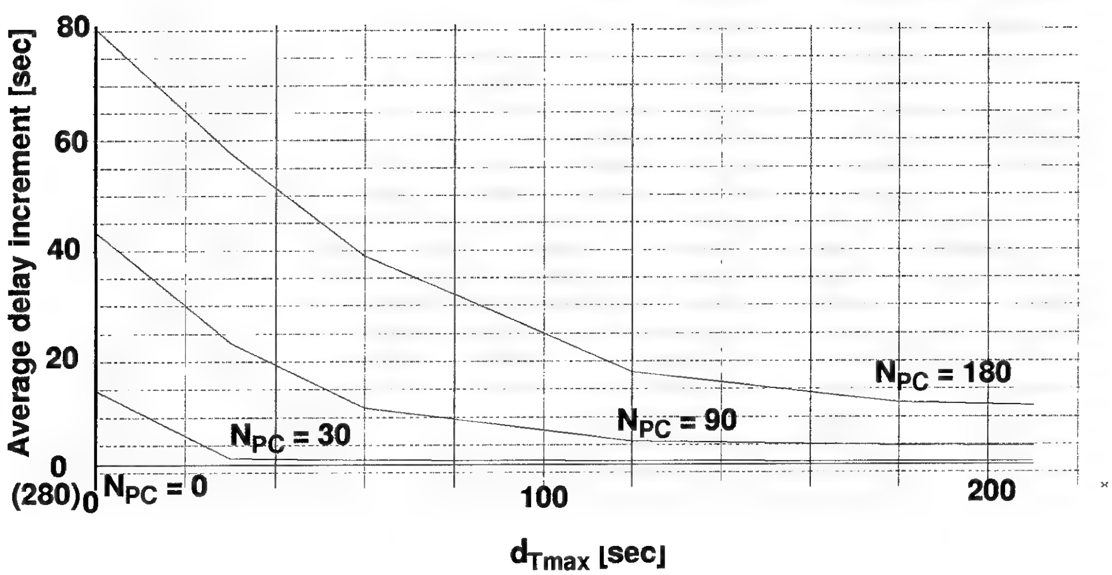


Figure 7.- Average delay increment for 36 aircraft/hour for a single runway vs d_{Tmax}

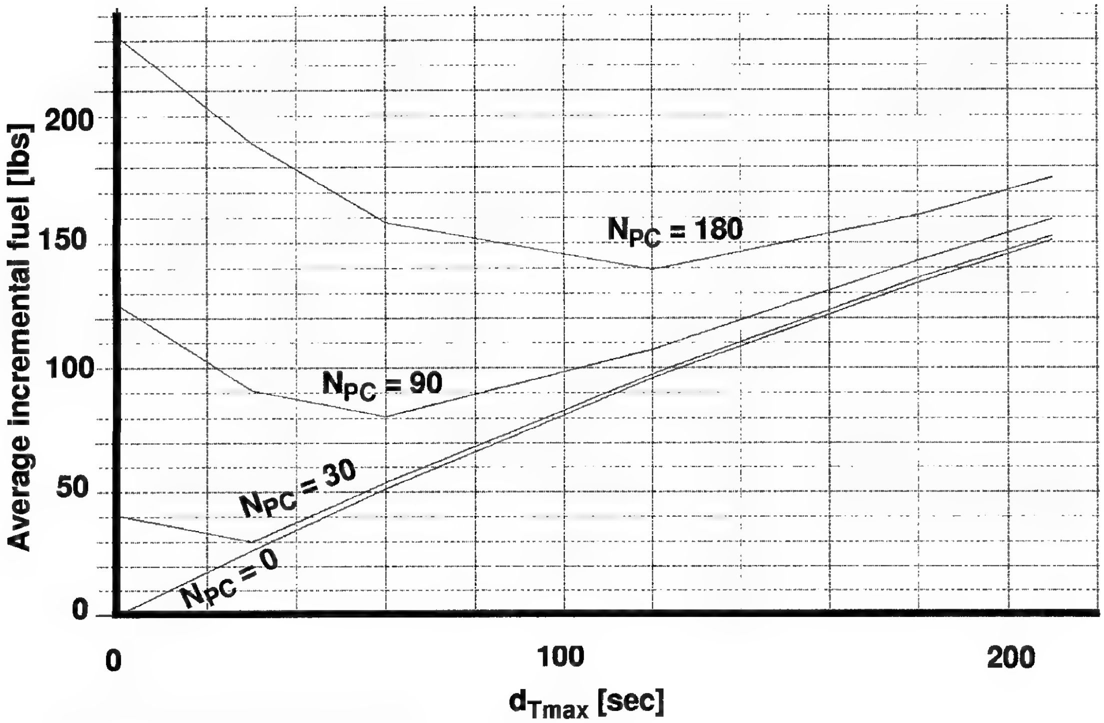


Figure 8.- Incremental fuel vs d_{Tmax}

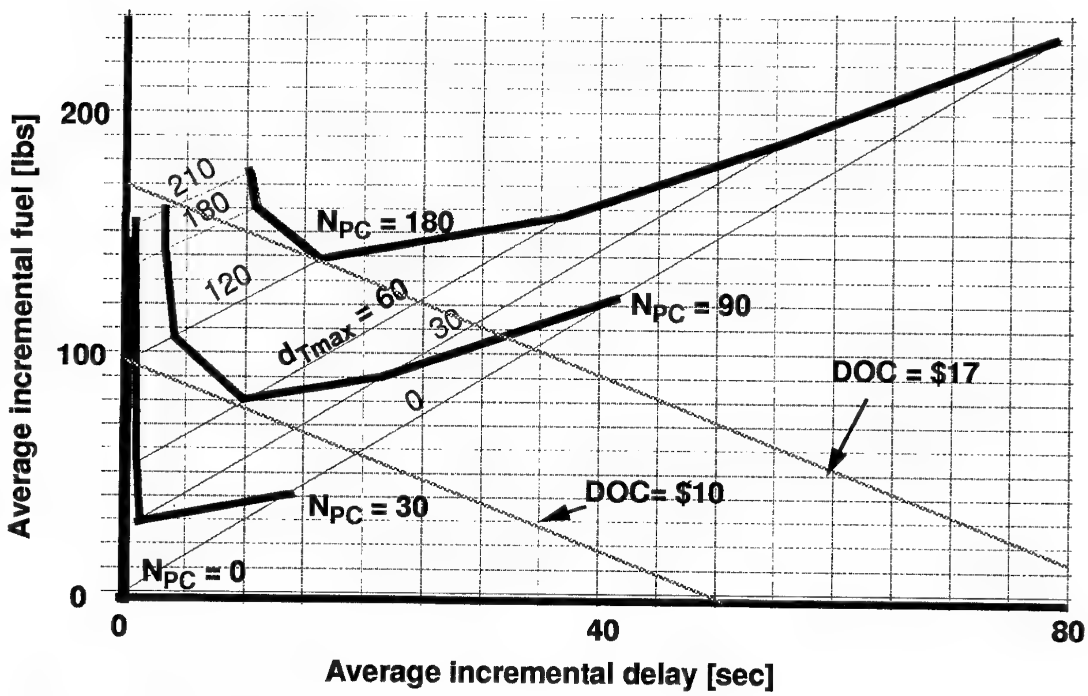


Figure 9.- Carpet plot

STRUCTURES, ARCHITECTURES AND DESIGN PRINCIPLES FOR DYNAMIC PLANNING FUNCTIONS IN ATM

U. Völckers
D. Böhme
Institute of Flight Guidance
German Aerospace Research Establishment
Lilienthalplatz 7, D-38108 Braunschweig
GERMANY

Summary

Air Traffic Management is a very complex and challenging domain. To cope with future traffic demand, while still maintaining or even increasing safety and efficiency of air traffic operations, intelligent machine functions have to be developed to assist the human operators in their mental control tasks. The specific requirements of the ATM domain necessitate sophisticated and well-designed assistance tools. Their most significant characteristics, design principles and structures are discussed and examplified in a real-world application.

1 Introduction

The approach to handle future air traffic with increasing traffic density, very complex air traffic management and control functions and an increasing demand to manage air traffic operations more efficiently and economically, is to support the human Air Traffic Control Operator by intelligent machine functions for *situation assessment*, *diagnosis*, *planning* and *decision-making*. Computers should not only do "simple" information processing but should often also take over human cognitive abilities. In many applications a desired behavior of the *world*[1] should be achieved by a "real-time plan-based control" (see e.g. /1/).

In general, automated planning requires a consideration of the *world's* state on a certain level of abstraction, which is referred to as *situation*. Planning can then be defined as the problem of generating a sequence of actions that transfers a *world's initial situation* into another *situation*, which satisfies the conditions of the planning goals. In case the *world's* behavior can be predicted restrictedly only with respect to time and accuracy, which is mostly the case when human operators are involved in managing[2] and executing plans, planning has to be done by using information feedback in order to monitor the compliance between the planned and actual *situations*. This is termed as *dynamic planning*.

Although it is the intention of this paper to present ideas regarding *dynamic planning* which are rather domain-independent[3] it is written with the background of work that has to be done for ATC automation /2,3/. Typical planning problems of the ATM domain and basic concepts and structures how to resolve these problems are sytematically discussed. A specific example which has been developed and elaborated, shows how these design principles have been successfully applied.

[1] the technical or environmental system which is controlled

[2] planning and/or controlling

[3] presupposing that the *world* is characterized by uncertainty, real-time constraints, and by involved human operators

2 Management and Control of Complex Systems Based on Dynamic Planning

Most complex systems can only be managed/controlled if future consequences of actions are taken into account. Skilled human operators normally try to predict the potential future systems behavior by using an internal model of the control task, whereby they are able to consider uncertain as well as incomplete information. But in unexpected strange situations, that require a fast reaction, the human operators interrupt their mental planning and focus their attention on the present control task to achieve a safe situation. This skillful human behavior suffers when the operator is over-loaded with information and the operator's information processing cannot follow system's dynamic.

Although a computer can process a much higher data volume, a technical copy of the human property of being capable of fast reaction (by switching the internal control structure) is also necessary as well as the ability to plan under uncertainty. For that purpose figure 1 shows a suitable, general, three-level architecture, which can be derived from a functional decomposition of a generic closed-loop decision making system /4/.

The system to be controlled is situated on the basic level. In the ATC domain it can be a certain traffic area (e.g. an airport) with an underlying structure (e.g. a network of taxiways and runways) containing a changing set of aircraft. On this level the dynamic behavior is described by a time-variant state vector (e.g. vector of aircraft positions, speeds, etc.)[4] and is affected by the environment (events, disturbances) as well as by (re-)actions (e.g. controller instructions and commands, given guidance signals).

[4] The state vector might contain qualitative information (e.g. weather conditions) that cannot be measured but that are observable by the human operator just as a part of the quantitative states (e.g. aircraft which can be seen by the tower controller looking through the windows).

Paper presented at the Mission Systems Panel of AGARD and the Consultant and Exchange Program of AGARD held in Madrid, Spain, on 6-7 November 1995; Châtillon, France, 9-10 November 1995; and NASA AMES Research Centre, California, USA, 16-17 November 1995 and published in LS-200.

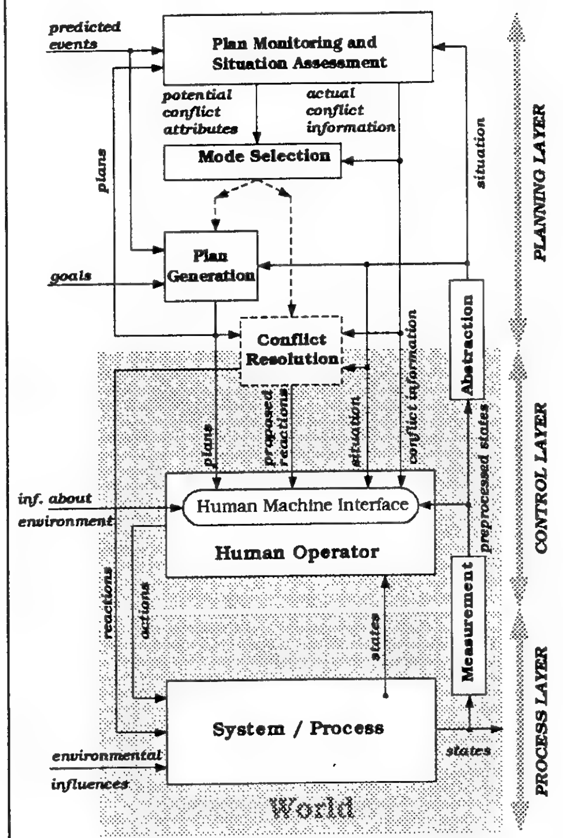


Fig. 1. A general architecture of an "intelligent", plan-based management and control

A computer-supported control requires a state measurement that usually does a low-level kind of information abstraction by suppressing fault sensor information or by calculating "artificial" states in case of lack of sensor information, in order to present full state information. Further decision support can be given by offering information about the environment that will have an influence on the system in future.

In order to realize a plan-based control further abstraction is necessary since planning can only be done in reasonable time if the model of the *world* has a suitable granulation. The result of this abstraction process is a time-variant *situation* that is assessed permanently[1] and that is explored in the plan generation process which is started under certain conditions. *Situation* information might be delivered to the human machine interface, too. *Situation assessment* is based on the results of the monitoring process, checking the conformance between planned and actual *situation*, but goes beyond that by exploring the "severeness" of detected present and future differences in order to give necessary information for a suitable mode selection. For such switching of control structure it is necessary to consider *situations*, information (predictions) about relevant future events, if available, but also previous plans. Thus a further information loop exists.

When an actual *situation* is assessed as "critical" since it causes an *actual conflict*, the human operator is provided with *conflict information* (e.g. aircraft which causes the conflict; involved aircraft; location). Under such circumstances a further decision support can be given by an automatic conflict resolution function[2] that immediately influences the system (e.g. by using guidance signals) and/or proposes the right sequence of (re)actions in order to resolve the situation. However, under most operational conditions actions can be drawn from plans.

It should be remarked, that such an hierarchical organized control should not imply a master-slave principle, since the human operator must have some means to have an influence on the planning according to his intentions /2/. This kind of information feedback is not shown in Fig. 1.

3 Planning Problems and Basic Concepts

In a common sense planning can be described as the task to determine the actual and future interactions with the *real world* to reach well defined goals. This can formally be expressed as the search for a transformation with *operators* or actions from a known *state* into another *state*, which satisfies the *goal descriptions*. It is clear that formal descriptions of *states* and *operators* are needed, too. These might lead to the well known "classic" planning problems, like the *frame problem* /5/ which have to be considered but will not be discussed in this paper.

In the domain of airport surface traffic management there are other important problems which are related to the required safety level. Because of the continuous traffic at major airports it is not practicable to alternate between planning and implementation of a whole plan. Instead planning and implementation must be interlaced. Of course, in this domain planning has to be based on a permanent plan monitoring to realize a closed information loop. The use of planning as an intelligent, look-ahead, closed-loop control (*reactive* or *dynamic planning* /6,7/) presupposes the solution of the following problems:

- ☐ planning under uncertainty, and
- ☐ planning under real-time demand.

These problems will be described in more detail in the next sections and also some general approaches will be given.

Of course, the design of the system as well as the accommodation of the present operational procedures to such a system have to be done in view of the controllers' acceptance and their workload. These aspects have an

1) The term "permanently" should be interpreted as " done highly frequent".

2) Automatic real-time conflict resolution is not necessary in all applications.

backward effect not only on the HMI[1] design but on the planning algorithms, too. Some questions about the controllers' ability to exercise influence on the planning and the proper plan representation are answered in another section.

Since *dynamic planning* with information feedback is not sufficient to fulfill all requirements, another important planning principle, that is termed as *recurrent planning with sliding horizon* is necessary which is covered in last section of this chapter.

3.1 Uncertainty

3.1.1 Reasons for Uncertainty

If consideration is given to what information is necessary to describe a *state*, the chosen set of measured signals from the *world* and their derived values to extract the relevant information must be listed. But, the actual available knowledge about

- ☐ the intentions of the *agents* (aircraft controlled by pilots) according to their plans (the taxi-instructions given by the controller),
- ☐ the predicted future events which will influence the behavior of the *world* (the inbound traffic etc.),
- ☐ the actual and known future operational conditions, constraints and goals

also belong to a description of a *state*. It should be noted that for all types of information there are different degrees of certainty. For instance, in spite of the unavoidable errors in measurement, the actual position of an aircraft is "better" known than the predicted touch-down time of an arrival. Beyond that, it is clear that the uncertainty of any predictions will increase with an expanding time-horizon.

Any planning of actions to act upon a certain *world* to reach planned or given *states* requires the calculation of variations of the system's behavior. Therefore a *dynamic world model* is needed which is calculable in fast time. For that and other reasons, such as the human involvement, the limited accuracy of measurement, etc., any model is not able to copy the dynamics of the real *world* without errors. Thus, the uncertainty is caused by (a more detailed description is given in /8/)

- ☐ the inaccurate assessment of *world* signals,
- ☐ the inaccurate and erroneous prediction of future events,
- ☐ the inaccurate modeling of the *world*.

Now some general approaches for planning under uncertainty are discussed, which are also used in the TARMAC[2] /9,10,25/ planning system, developed by the German DLR.

3.1.2 Planning with Time Intervals

If the *world model* is characterized by continuous time variables it is often useful to use time intervals for planning, which was introduced by Allen /10/). This allows to include the uncertainty of the prediction about the exact time of an event in the planning process. The size of the time interval, within which the predicted event will happen, can usually be determined according to on-line statistics or the probability density functions of estimated model parameters (to predict an event). Otherwise a "fit" size has to be assumed "per definition" (this problem is discussed later).

If events are related to time intervals, then a *world model* is needed which is able to extrapolate the intervals into the future. This can be explained with the example of the taxi-path planning for an arrival. The event "landing" is marked with a time interval $I = [t_1, t_2]$, with $t_1 < t_2$. Within I lies the assumed/predicted touch-down time. For any calculation of a possible taxi-path the aircraft motion model has to be applied to the times t_1 and t_2. Thus, for a particular location *A* on the airport an other interval $I_A = [t_{1A}, t_{2A}]$ can be calculated, with $t_{1A} < t_{2A}$, $t_1 < t_{1A}$, and $t_2 < t_{2A}$. In general this leads to two possible views (fig. 2):

- ☐ a resource is always used by/allocated to an agent for a certain duration (at every certain location the aircraft may be there for a certain duration),
- ☐ at every certain time an agent may use several resources (at every certain time an aircraft may potentially be at several locations).

If furthermore the inaccuracy of the model is taken into account, the sizes of the intervals will increase with time according to the growing uncertainty (lowest and highest expected speed of an aircraft). A better interpretation can be obtained if the durations[3] for the use of a resource according to the *dynamic world model* are extended through *buffer intervals* (against uncertainty) on both sides (fig. 2). Such an extended interval shall be called *occupancy interval* (I).

[1] Human-Machine Interface

[2] Taxi And Ramp Management And Control

[3] which are needed by an aircraft to pass certain taxiway sections of a certain taxi path

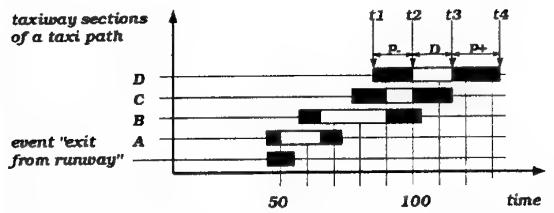


Fig. 2. *Occupancy intervals* and their expansion caused by the increasing uncertainty

The figure shows the extension of the planned resource allocation time intervals and their expansion caused by the increasing uncertainty illustrated for the example of the taxi-path planning. Let the taxi path of an aircraft be *A-B-C-D-...* , where *A, B, C, D,...* are the consecutive taxiway sections of the taxi path; D_A, D_B, D_C, D_D, ... the corresponding durations (for this aircraft); t_0 is the present time, $t_S = 50$ the predicted time when the aircraft will start its taxi operations (for instance the predicted runway-exit time); $t_P \leq t_0$ the time when the prediction was made, and $\varepsilon_0 = f(t_S - t_P)$ the (assumed or calculated) uncertainty interval for this prediction.

Each interval **I** calculated in such a way is described by

$$\mathbf{I} = [P_-,\ D,\ P_+] = [t_1,\ t_2,\ t_3,\ t_4],$$

where

$$P_- = [t_1,\ t_2] = \left[t_2 - \varepsilon_1 * (t_2 - t_S) - \frac{\varepsilon_0}{2},\ t_2\right]$$

and

$$P_+ = [t_3,\ t_4] = \left[t_3,\ t_3 + \varepsilon_1 * (t_3 - t_S) + \frac{\varepsilon_0}{2}\right],$$

are *buffer intervals* and $D = [t_2,\ t_3]$ the duration of the resource allocation. It should be noted that the *prevention intervals* $P_\pm$ increase with time. Furthermore it is important that the size of the *prevention intervals* depend on the uncertainty of event prediction ε_0 and the parameter ε_1.

However, planning of an occupancy of a certain resource for an interval **I** according to the known/assumed uncertainties leads to two additional problems which have to be solved:

- □ The ε-Parameter Tuning Problem
- □ The Conflict Evaluation Problem

3.1.2.1 The ϵ-Parameter Tuning Problem

The greater the prevention against uncertainty (larger ε),

- □ the higher is the "stability" of a *plan* (that means: the lower the "probability" that the *plan* becomes inadequate with time), but also,
- □ the smaller is the set of possible/permissible *plans*, and
- □ the less good is the *optimal plan*.

As the values of the suitable ϵ-parameters cannot be obtained by theoretical considerations in general, they have to be tuned in the real application or better: with the help of a *world* simulation.

3.1.2.2 The Conflict Evaluation Problem

During the planning process it is assumed that every *agent* allocates some resources over certain time intervals (every aircraft occupies certain taxiway sections for certain time intervals). Therefore a *planning conflict* can be defined as the use of a certain resource by two or more *agents* at the same time. That means there is at least one pair of agents which have overlapping intervals $\mathbf{I}_{Aa}$ and $\mathbf{I}_{Ab}$, where A names the resource (location), and a and b index the agent (aircraft). Except some specific cases, such as deadlocks, an overlap does not necessarily lead to a real conflict, even if no unforeseen events will happen.

Planning with time intervals requires a conflict trust evaluation[1] as a function $f_c = (\mathbf{I}_{.a}, \mathbf{I}_{.b}, t_C - t_0)$ of the two intervals and the time difference between the earliest conflict time t_C and the current time t_0. Only in case the value of f_c exceeds a fixed threshold, the currently calculated plan is rejected. The measure f_c should have the following properties (fig. 3):

- □ The farther the *conflict* lies in the future, the lower is the trust that the *conflict* will really happen.
- □ The more the intervals $\mathbf{I}_{.a}$ and $\mathbf{I}_{.b}$, and also the corresponding duration intervals D_a and D_b overlap each other, the greater is the belief.

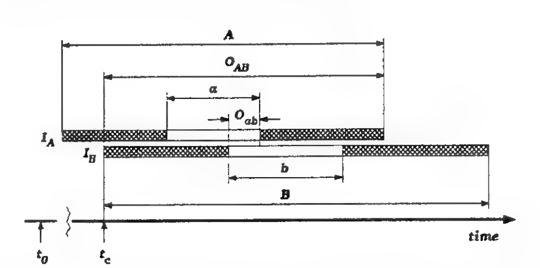


Fig. 3. A planning conflict represented by overlapping time intervals **I.** and its reference to the measure f_c.

f_c can be determined by:

$$f_c = \left(\frac{\left(\frac{\mathrm{O_{a,b}}}{\mathrm{min(a,b)}}\right)^2 + \left(\frac{\mathrm{O_{A,B}}}{\mathrm{min(A,B)}}\right)^2}{2} \right) * \left(1 - \left(1 - \exp(-(\mathrm{t_c} - \mathrm{t_0})/\mathrm{T})\right)^{\mathrm{n}}\right)$$

$$T > 0,\ n > 0$$

where A, B are the interval sizes of $\mathbf{I}_{.a}$ and $\mathbf{I}_{.b}$; a, b are the sizes of the duration intervals D_a and D_b; $O_{a,b}$, $O_{A,B}$ are the amounts of the overlap of **I**- respectively D-intervals; and T is a proper time constant.

---

[1] similar to a fuzzy predicate /11/

3.1.3 Plan Monitoring and Situation Assessment

3.1.3.1 Basic Considerations

Any *reactive planning* has to be based on a permanent or repetitive /7/ comparison between the *state* of the *real* and the *planned world* at the present time. This can be called *plan monitoring*.

First it should be noted that not every difference between the planned and the real *state* is discernible, because

- there is always a limited accuracy of the measurement of the *world* signals, and
- there might be a certain granulation of the *world model*, which is used to describe the planned *state*.

However, even a recognized difference between the *state* of the planned and the real *world* does not inevitably mean that re-planning is necessary. For the judgement whether planning should be done, the planning system must contain a "look-ahead unit" /12/ which extrapolates the *actual world* into the future considering the remaining *operators (acts)* of the present plan. This can be done better one abstraction level higher than the *state-level* - at the so called *situation-level*. Therefore any *situation* assessment requires the prediction of the future *states* of the real *world*.

In the ATC domain usually an *actual conflict* is caused by a pilot violating his/her plan, that means he or she does not follow the instructions of the controller. One example is the deviation of an aircraft from the instructed taxi path. A more general consideration of an *actual conflict* must include all circumstances that require an immediate action by controllers (commands, warnings etc.) and pilots of the certain aircraft or the set of involved aircraft. The detection of such rare conflicts is of great importance, but not the only task of *situation assessment*.

Under "normal" operational conditions one has to expect a frequent non-conformity between planned and actual *situations*[1]. However, even in case of non-conformity adaptation of plans is not necessary as long as it is or seems to be sure that neither the *world* will be led into a critical situation nor any goals will be lost. But often the *situation* should be assessed as a *potential conflict*, that means there is a certain chance that planning is or will be necessary in the near future in order to generate proposals for suitable actions.

Since there is no objective "belief measure" and hence it is not obvious whether planning process should be started, and if so, when it should be started (immediately or within a future time interval), there is another objective of *situation assessment* to give some clues to the plan generating unit to answer these questions.

So, besides detection of *actual conflicts*, a suitable evaluation of *potential conflicts* is required.

The result of the *plan monitoring* and especially of the evaluation of a detected difference between the planned and the real *world* should be a classification of three categories:

- The difference is tolerable that means no *conflicts* were detected.
- The present *situation* requires a replanning immediately or at a later time (this point is viewed in the following section).
- The *situation* is crucial (runway incursions, deviations) in the sense that it requires an immediate reaction by the system (guidance signals) and/or by the controller (commands to the involved pilots). So, first of all the present *situation of the world* has to be transferred into a safe *situation* without planning which then later will allow a normal time-consuming planning.

The evaluation of possible future conflicts through the *situation* assessment should be based on the same considerations as pointed out above. Finally it should be mentioned that the monitoring task can easily be decomposed and distributed to several monitoring processes (units) corresponding to the involved aircraft and/or specific topological elements (taxiways, junctions, areas etc.).

3.1.3.2 Detection and Evaluation of Conflicts

Actual Conflict Detection

When an *actual conflict* detection function for a certain application area is to be realized many specific requirements have to be considered. In the ATC domain, and especially for future SMGC[2], which will be based on planning of the push-back times and schedules, taxi path etc., there are two approaches, that differ in what is supervised: aircraft or areas.

The first one – aircraft monitoring – requires the monitoring of each aircraft whether taxi operations are done in conformance with its plan, especially with the planned taxi path, since every deviation has to be assessed as an actual conflict. A plan-based short- or medium-term prediction of further movement[3] has to be done additionally to detect threatening collisions of aircraft as well as dead-lock situations.

Following the second approach, an *area monitoring process* has to be informed of how incoming aircraft have to operate within and how they have to leave the certain area in order to have necessary information for conflict detection. Also the consideration of "measurable" obstacles with respect to actual conflicts can be done easier here than by aircraft-focused conflict detection.

[1] In many cases the planning process does not calculate future (planned) *situations*, but a sequence of actions. Giving a certain *initial situation*, a planned sequence of actions, and the actual *situation*, the monitoring process is able to detect whether the actual behavior of the *world* differs from the planned one or not.

[2] Surface Movement Guidance and Control

[3] This should be termed more exactly as "pilot aircraft control prediction", because prediction also has to consider pilots behavior, especially the adaptation of aircraft speed to the observable traffic situation.

Taxiway junctions, intersections and sections[1)] are basic elements of a *monitoring area*. *Monitoring areas* might overlap, but coverage of the whole traffic area should be guaranteed.

For the explanation of conflict detection done by an *area monitoring process* an example might be helpful. Assume that a *monitoring area* consists of only one taxiway junction (intersection) and all taxiway sections leading to this junction. Then conflict detection can be based on the following information about the behavior of each aircraft

- ☐ the taxiway section the aircraft will use when outgoing
- ☐ whether, and if so, under which condition the aircraft has to stop before crossing and its pilot is allowed to continue taxiing with or without controller's command (at a certain time or after an event that is observable by the pilot, like crossing of another aircraft) and also
- ☐ the right-of-way or crossing sequence.

Both approaches are certainly exchangeable however, they offer a useful redundancy with respect to reliability.

Qualitative and Quantitative Evaluation Based on Conflict Attributes

Detection and a following evaluation of a *potential conflict* requires a prediction of future aircraft movements as well as of future events causing a change of constraints. Such a prediction is inevitably uncertain for many reasons /2/, that should be considered by the calculation of possible future *situations*, but is basis of the recognition and evaluation of two types of *potential conflicts*:

1) *Potential collision conflict*: An aircraft may violate future constraints, especially those which will be caused by other moving aircraft.
2) *Potential goal conflict*: An aircraft may not satisfy the planning goals, that means for example: it may not arrive at its destination within the required time interval.

Contemplating the SMGC domain again, aircraft will occupy all sections of their taxi path over a certain time, which can be calculated with the help of corresponding average aircraft movement times. In section 3.1.2 an extension of these "origin" occupancy intervals by uncertainty intervals was supposed. Thus the resulting *occupancy intervals* of sections, that follow subsequently in the taxi path, overlap each other in every case. Assuming that aircraft movements will be executed over all taxi path sections within the corresponding *occupancy intervals*, a *potential collision conflict* requires necessarily that *occupancy intervals* of distinct aircraft overlap each other. However, in order to assess a superposition of intervals as a *potential collision conflict* several aspects have to be considered, especially

- ☐ the degree of superposition and
- ☐ the remaining time up to the potential collision

as well as qualitative aspects, that result from the predicted site of the potential collision[2)] or a detected potential dead-lock situation. The evaluation of *potential goal conflicts* can be done in a similar way.

The interference of all quantitative aspects might be measured by only one special defined function, but a more general approach is the use of separately evaluated conflict attributes, like severity, urgency and duration as introduced in /13/. Although in this case for each quantitative attribute a suitable measurement function is needed too, it is much easier to define these functions. Furthermore there are two additional advantages: It is not only possible to incorporate qualitative attributes, but also if necessary a stepwise change of the attributes set during the development process. The "degree of threat" can be derived through an "evaluation logic" using a set of (fuzzy) rules to process the values of all attributes and only if it exceeds a certain threshold the situation must really be regarded as a potential conflict.

3.1.3.3 Conflict Evaluation and Planning Mode

The detection of *actual conflicts* and the quality of the corresponding (automated) conflict resolution is essential for the degree of safety that can be performed by a computer-supported control, whilst evaluation of *potential conflicts* can help to accomplish the necessary plan stability (see e.g. /14/), that means:

- ☐ the plan changing rate should be as small as possible and
- ☐ updated plans, that follow subsequently, should be as "similar" as possible.

Such a "conservative" behavior of the dynamic planning is necessary to get the controllers' acceptance, since otherwise his workload would increase.

One approach to achieve plan stability is based on a three-layered plan generation process, whereby in each case the selected layer, that generates a new or modifies an existing plan, can be called a *planning mode*. The three *planning modes* (types of planning) can be described as follows in a descending order /2/:

1) *New planning* or *repeated planning* without any information about an old plan, since such information is either not available (e.g. for a new incoming aircraft) or not useful
2) *Replanning*, which means planning under consideration of old plan information
3) *Plan modification* in which only one item (specification of an action, e.g. the push-back time) of a plan is adapted

---

1) Here "taxiway" is used as synonym for runway.

2) For reason of safety at least it should be distinguished between taxiway-taxiway and taxiway-runway crossings.

Plan generation done by the several layers requires not only different efforts but also produces plans which differ in the similarity to the old plan, if there exists one. However, plan generation of a certain layer may not succeed, that means no suitable plan can be found. In this case the process must be repeated on a higher layer. In order to start plan generation already on the appropriate layer, *conflict attributes* provided by *situation assessment* are doubtless necessary. Considering the circumstance that several aircraft may be involved in a *potential conflict* and therefore several plans possibly have to be changed, the design of a "planning mode selection algorithm" is a generally very difficult and unsolved problem.

3.1.3.4 Determination of the Planning Necessity

The correlation between an increasing *planning horizon*[1] and the increasing uncertainty already has been explained. This bears the thought that automatic planning should be done as late as possible to limit the *planning horizon* as much as feasible. "As late as possible" means there is enough time to compute a sufficient plan, but also enough time for the involved humans to accept/understand and possibly to transmit the plan.

But there is a second aspect, which also has an influence on the determination of the *planning necessity*, and which relates less to uncertainty but rather to the quality of the best plan. If the plans for the agents (aircraft) are made in the same order as the agents become known (or have to be replanned as the result of *plan monitoring*), the limited resources are assigned to the agents in the same order, too. This "first-come-first-served" method is equivalent to a non influenceable ranking of the agents involved in the planning process. Therefore a skillful ordering might influence the quality of the plans by grouping some agents and computing their plans at the same time and/or (if the planning task becomes for complex) by determining a subset of aircraft which should be planned first.

3.1.3.5 Situation Assessment in Real-Time

As explained in the previous section, planning functions are based on situation assessment working permanently. So it is clear that it has to be done rapidly in order to achieve a highly frequent recurrence. Since detection of actual conflicts is one objective, situation assessment has not only to be performed fast but under well defined real-time conditions.

A very promising approach to this need is the decomposition of situation assessment in sub-tasks which then can be executed in parallel if a network of computers is available. Beyond that, this possibly bears the advantage of being a rather fail-safe solution. Decomposition of situation assessment should utilize the circumstance that monitoring can be focussed on aircraft as well as on areas.

Figure 4 shows a system performing parallel situation assessment that consist of

- ☐ a variable set of aircraft monitoring processes A_i,
- ☐ a fixed set of area monitoring processes N_k,
- ☐ a supervisor process, that controls aircraft and area monitoring processes and performs conflict detection (D) as well as conflict evaluation (E),
- ☐ a knowledge base that serves as a data interface, too.

Each process A_i observes the movement of the aircraft i and predicts its occupancy intervals for all sections of its taxi path that are delivered to the potential conflict evaluation process E. Furthermore it proves whether the aircraft is still able to execute the next planned action, for example: whether the aircraft is able to stop (as planned) or is able to turn into the required taxi way etc. Such information of short-time prediction is given to the actual conflict detection process D.

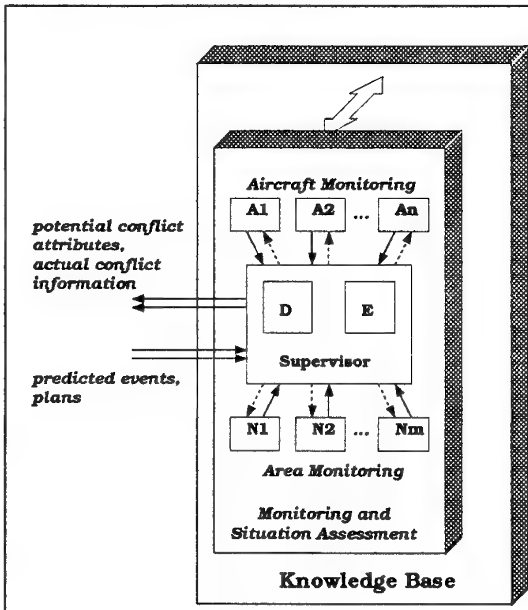


Fig. 4. Architecture for real-time situation assessment

Every process N_k monitors the compliance of the planned taxi paths and crossing order(s)[2] of all aircraft moving on area k. Deviations of aircraft from the planned taxi path are always actual conflicts, whereby predicted changes of the crossing orders are assessed as potential conflicts since they might be tolerable with respect to the planning goals.

3.2 Real-Time Demand

Real-time demand leads to the most difficult problems in the field of automatic planning. Although a lot of work has been done under several perspectives it seems there is no general solution especially for complex

[1] *Horizon* means depending on the context either the farthest future time, up to which is planned, or the duration from the present to this time.

[2] an area k may contain several intersections

domains like ours. The main thought which was dealt with was to reduce the complexity of the planning task and/or to speed up the planning process. We will also consider both points on an abstract level. However, since fast-time is not "real-time" it should be viewed what "real-time demand" means in the context of direct control.

Control in complex systems is in general hierarchically organized. From the lowest layer up to the highest layer the degree of abstraction rises as well as the considered time horizon. The specific task of each layer has to be solved under given constraints by the superior one /16/.

For automated direct or tactical control at the lowest layer (e.g. in the aircraft control systems) a common agreement exists about the meaning of the term "real-time". The control task (value computing of the control variables, handling of exceptional cases, etc.) has to be finished within a certain period. An appropriate duration of the period can usually be assessed by analyzing the system's dynamics. Since the computing normally is simple even for highly sophisticated control algorithms, a worst case study of the time consumption can be done. The quality of control is often determinable in advance with analytical methods, because it can be assumed that the input *world* signals stay nearly constant during a period.

However, these statements cannot be applied directly to planning tasks on higher layers. The most important reasons for this are:

- ☐ An appropriate duration of a period in which any planning process should terminate cannot be fixed a-priori. Only the latest time can nearly be settled (regarding to a corresponding future event) when the planning process should finish.
- ☐ During a planning process the *world* cannot be assumed as invariable. Even cases have to be covered, where an ongoing planning has to be cancelled automatically and restarted, because the start conditions are no longer valid (*non-monotonic planning*).
- ☐ Planning is a very complex task which incorporates several sub-tasks, such as *plan monitoring*, determination of the *plan necessity time*, *single planning*, etc., which have different time demand.
- ☐ Sometimes humans are involved in the planning process.

This is a much worse situation than in the context of direct control. Normally the treatment of real-time planning starts from the following underlying assumption: If the planning process needed no time, no problems would occur. Therefore the less time the planning process consumes the less problems have to be solved, or the easier the problems can be solved respectively. Following this assumption the main questions are:

- ☐ How can the planning process be speeded up?
- ☐ How fast must the planning process run to ensure that the time demand problems can be solved ?

Of course, these questions lead to more, still unsolved problems. And there is no doubt that they can only be answered with the help of a comfortable *world*-simulation system.

3.2.1 Speed-up of the Planning Process

There are two main options which might be useful to speed up the planning process.

1) Whenever possible the planning task should be decomposed and be distributed to different planners. This can be done horizontally or vertically.

   - ☐ A horizontal decomposition means to split the planning tasks into smaller, easier and faster solvable tasks so that the aggregation of all partial plans solves the original planning problem. The decomposition should be oriented on the natural structure of the *world* for instance, different planners for the different aircraft or different areas of an airport. Of course, by doing this, a lot of difficult unsolved problems have to be solved, which are related to such fields like *distributed planning*, *multi-agent planning*, and *cooperative planning* /17/.
   - ☐ If the decomposition is realized in such a way that there are several layers of abstraction /18/ it is called "vertical" and leads to the concept of *hierarchical planning* /19/[1] which is similar to the multi-layer control structure mentioned above. The planning of a sequence of *world situations* without (a detailed) conclusion on the actions is a special (but in the ATM-domain very useful and already practiced) variant of a vertical decomposed planning. In the subordinate layer of such a *situation planning* the experiences of the human-operator can be used for the determination of an appropriate sequence of actions (control instructions for the pilots) to achieve the desired *situations*.

2) The characteristic of the second option is the reduction of the developmental possibilities through reduction or limitation of the *planning horizon*. This can be done either by generation of sub-goals, the execution of which guarantees or facilitates the satisfaction of the original goal, or by fixing of a certain limited *horizon* (*incremental planning*) /20/. If planning of future traffic handling is still done by the controller and only supported by a *planning support system* there is in addition the possibility that the *horizon* can also be chosen by the controller according to the available time for the planning process or according to his assessment of the final planned traffic situation.

---

[1] This term is also used for a special architecture of the *distributed planning* of the TARMAC planning system where several, single-aircraft planners are coordinated by a superior planning unit).

3.2.2 Development Steps

In spite of the various methods to speed up the planning process (in doing so, all steps, which can be done on the level of implementation of planning algorithms into a computer network, must not be ignored) the question has to be answered how the time and speed requirements can be determined. It is believed that the development of every planning system working under real-time demand should be subdivided into four steps which are characterized by the "control of time" in an external *world* simulation:

1) The basic algorithms of the planning system (respectively subsystems) should be tested on the basis of a set of singular situations obtained through "flashlight photos" of the *world*. Singular situations in this context means that the reactions of the planning system for one situation never do influence another situation. Between all situations there are no time relations. Therefore no time modeling in the external simulation is needed.

2) In this step the simulated *world* time is increased only if the whole planning system (all subsystems/units) has finished its work. The behavior of the *simulated world* now depends on the start situation as well as on the *recurrent planning*. Since the simulation runs in a triggered mode an infinite fast time planning system can be modeled and no real-time problems occur. But, since it is difficult to incorporate humans in such a simulation their behavior has to be modeled, too.

3) In this step the time of the *world* simulation is independent of the planning system but the time runs n times slower than real-time, were n is a time extending factor which is set by the system developer. Thus a planning system can be tested which runs as fast as desired. If the test starts with large n (to simulate a nearly infinite fast planning system) and n is made smaller step by step then the occurring real-time problems will become more and more difficult and the results of the planning system will impair, too. Then there is a certain n for which the planning systems work is insufficient according to the requirements. Therefore, the question can be answered either how fast the planning system should be at least or a reconstruction (architecture and/or algorithms) of the planning system is necessary if no realistic chance is seen to fulfill the time requirement.

4) Of course, in a final step the results have to be evaluated using an implemented planning system and the humans involved. This requires a real-time simulation of the *world* in which the controller can work under realistic working conditions, or a field implementation.

3.3 Human-Machine Interaction

The problem of human-machine interaction is often reduced to the problem of human-machine interface design to answer the question how the human-operators can work with the system. Of course, the layout of the interface is very important for the human-operators' acceptance, because it has a direct effect on their workload. However, in the context of *automatic planning*[1)] there are many other questions which have to be answered long before even a prototype of an interface can be made. These answers react upon the implemented planning algorithms and the used planning methods.

So, how does a *reactive planning*, the plans of which are made for a human-operator, differ from others which are made to control machines, e.g. robots? Focussing our attention on planning for humans, especially for controllers, the main differences are:

- ☐ There is an general guideline that the controllers retain the authority in such a human-machine system as well as they should keep the responsibility for traffic handling. Both points presuppose a possibility to influence the planning system.
- ☐ For the controllers it should be possible to realize the plans without an increasing workload, therefore:
  - ☐ *Single-plans* must not be changed as often as it might be desirable to achieve the planning goal in an optimal way (shortest time, least expense, etc.).
  - ☐ If it is unavoidable to change a *single-plan* it should be done in such a way that the new plan is "similar" to the old one. This problem is called the *plan stabilization problem*.

The general problem of the role of a human-operator in a human-machine system should not be discussed in this paper /21/. However, if only the technical side is discussed, the question remains how a controller could influence an automatic planning system. Two ways of the controllers influence on an automatic planning system should be distinguished: the *direct* and the *indirect influence*.

The *direct influence* is characterized by the controller modifying or replacing a calculated plan by his own one. An often discussed example is the ability of the controller to assign a new (in his mind better) taxi-path to an aircraft. But, as already pointed out, the plan of the controller is needed for *plan monitoring* and for further planning. Therefore the controller has to "inform" the planning system about his plan. Although there are modern communication methods (window surfaces, voice recognition etc.), which enable him to do this with little effort, a crucial problem is hidden - the difference between the controllers and the machines *world model*. There might be cases where only the

---

1) It is important to keep in mind that the human-machine interaction is only pointed out for the case that there is an **automatic** planning system, not an interactive one.

planning system detects a *planning conflict*. Then the planning system would have to ask the controller how he would solve it (e.g. the future right of way order for the aircraft at a certain junction). Depending on the answer of the controller, the planning system can have the "impression" that plans of other aircraft have to be modified according to the controller's intention. So in certain cases, either a very complicated time consuming human-machine dialog has to take place or further planning has to be done under the uncertainty of an old *single-plan* that unavoidably leads to a loss of optimality.

If the controller has mainly the possibility to change planning conditions, constraints or the optimization criterion for traffic handling in the near future according to his intentions, he has an *indirect influence* on the planning process. Of course a translation layer as part of the intelligent HMI is needed which enables the controller to do this without any knowledge about the details of the planning algorithm. Examples on how influence can be exercised, are:

- ☐ the aircraft should not stop at certain locations,
- ☐ the plans should change less frequently, and
- ☐ a certain aircraft should have the highest priority.

The advantages in contrast to the *direct influence* are, that

- ☐ there is no time consuming dialog needed,
- ☐ the planning system has full information about all plans (consistency of information), and
- ☐ the controller is able to put influence on all future planning according to his intentions.

It should be mentioned that there are many additional difficult problems in the design of the HMI which are closely related to the planning algorithms, for instance, the timely display of planned actions (*plan translation problem*), and the determination whether (and if so, what) actions are necessary to reach the (next) planned *situation*[1)] (this is part of the *plan transformation problem*). But, since they do not feed-back to the design of planning algorithms they are not described here in more detail.

Now the second point is again considered, namely: what has to be done to enable the controller to realize the plans without an increasing workload. To guarantee that the plans do not change very frequently a plan with a sufficient prevention against the uncertainty is needed. For *plan stabilization* information of an old plan must be used for planning. Therefore it is expedient to introduce three types of planning:

- ☐ *new planning* or *repeated planning* without any information of an old plan (e.g. for new incoming aircraft),
- ☐ *replanning*, which means planning under consideration of old plan information either for constraints or for the measure of the similarity between a currently computed and the old plan (e.g. the planning of taxi operations on a former planned taxi path for an aircraft),
- ☐ *plan modification* in which only one item (e.g. the push-back time) of a *single-plan* is adapted.

In general the last two planning types are not only useful to change plans smoothly, but also to reduce the computation time for a plan.

3.4 Event-Driven Recurrent Planning with Sliding Horizon

Planning necessity results either

- ☐ from the predicted events of new agents coming into the *world* , or
- ☐ from the intolerable discrepancy between the real and the planned *world* which is detected by *plan monitoring*.

Since the second case can also be termed as *event*, one can be distinguish between

- ☐ *external events* which occur outside (mostly independently) of the planning system (e.g. the announcement of an incoming aircraft by ATC- or airport-systems; the information that a certain planning constraint will be changed), and
- ☐ *internal events* which are generated by the planning system itself (e.g. the achievement of any earlier determined time of planning necessity).

Subsets of the *external and internal events* initiate the planning immediately (*initial events*). In this sense planning is "*event* driven" and not subject to a certain time cycle.

If an *initial event* occurs, computation of new plans or modification of the corresponding precomputed plans is needed only for a small number (often only one) of agents. The majority of the other *single-plans* is not affected (a plan, which belongs to a certain agent, is called a *single-plan*). The *overall-plan*, which is the set of all *single-plans*, is modified through addition of some new *single-plans* or through modification of some existing *single-plans*. As in general the plans have to be calculated in advance, the planned *actions* or *states* for the near future do not change. But with every new planning process the *planning horizon* slides farther into the future (fig. 5). This principle can be called *recurrent planning with sliding horizon* (according to a similar concept in the field of optimal control; see also /15/), which is a special variant of the *reactive planning*.

Every new planning process starts with the updated information about the *real world state*. By using this information the uncertainty about the world's behavior decreases in comparison to the foregoing planning if only the time up to the *horizon* of that planning is considered.

[1)] In this context only plans were considered, which do not contain a sequence of actions but a sequence of desired situations.

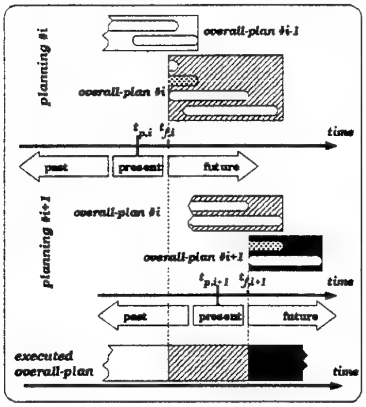


Fig. 5. Principle of the *recurrent planning with sliding horizon*

The picture shows two consecutive plannings (numbered by i and $i+1$). Planning i is finished at the time $t_{p,i}$ and was made under the constraints of the overall plan *#i-1* (and also might be constrained through earlier plannings). Since the planning reacts in advance the plan *#i-1* is not modified for the near future. This means that neither new single plans are added nor old single plans are modified. At the time $t_{p,i+1}$, when the next planning has finished, the whole procedure is repeated. If it is viewed from any time backward, it can be seen (below) that the executed overall-plan is compounded of the origins of the consecutive, earlier overall-plans.

4 Example: Runway Occupancy Planning for Departures

4.1 Problem Description

4.1.1 Objectives of Occupancy Planning

Although "capacity of an airport" is a commonly used term in the domain of Air Traffic Management (ATM), its exact definition is still disputable. The main reason for that seems to be the dependence of the capacity on various "environmental" factors, such as weather situation (wind, visibility) and traffic mix. Of course, capacity is primarily based on the runway system and the prevailing operational procedures, like the runway configuration. Many of these factors cannot be measured quantitatively but can only be described qualitatively. Furthermore some factors vary in time, thus capacity is time-variable, too /22/.

However, it is clear: airport capacity limits the number of movements and in case traffic demand exceeds capacity for a certain time (arrival peaks) the delays of the arriving aircraft grow. Therefore in practice, capacity is often regarded as the maximum throughput or its definition can be based on the throughput at which the average delay does not exceed a certain amount /16/. Since arrivals have to be served with priority, departing aircraft often do not reach their scheduled times, too. But the present operational procedures force a surprising interdependence between delays and capacity: On the one hand airport overloading "produces" delays, but on the other hand certain departure delays are necessary to reach maximum capacity in case the same runway is used for arrivals as well as for departures. This can be explained as follows: In order to fill any gap in the arrival stream with departures the controller, who has no complete information about future events, about the current traffic situation on the apron etc., needs a queue of departures waiting (with running engines!) at the departure runway to react (skilfully!) so that at least no constraints which are critical for the safety are violated.

Thus the main objective for an occupancy planning system is the enhancement of airport capacity by reducing the average departure delay. Moreover runway occupancy planning shall help

- to deliver the departures on time or within desired time intervals (slots)
- to coordinate the ground handling procedures of the airlines, the apron activities, especially the push-back sequences, and the superior departure planning

and in the farther future

- to coordinate arrival and departure management
- to generate planning goals for other planning subsystems of TARMAC.

In the domain of ATM any planning system has to be designed in such a way that

- controllers and pilots have to maintain the responsibility for traffic management and safety
- no increase of workload for controllers or pilots is acceptable.

Especially the latter aspect is not only important for the design of the Human Machine Interface but also has to be considered in the design of the planning algorithms.

In the following text runway occupancy planning will be restricted to departures.

4.1.2 Present Operational Procedure (OP)

Before an aircraft is able to depart, several activities are necessary which are done under the responsibility of four institutions: the airline itself, Apron Control, Tower Control, and Air Traffic Control (ATC).

The duration of the airlines' ground handling procedures determines the earliest time the aircraft could leave its parking position. The experience and knowledge of the pilots, their familiarity with the operational procedures of the airport and their own intentions lead to great variations of the times needed for push-back and taxiing.

Apron Control done by several apron controllers manages the traffic on the apron(s) of an airport. That means for departures: to schedule their push-back operations, to determine their taxi paths from gate to the movement area, and to give right of way instructions to the pilots.

Tower Control done by a controller named PL is responsible for taxi guidance, for runway management (line-up and take-off clearances), and to maintain

defined separations between aircraft on the standard (instrument) departures routes (SID) aircraft fly after take-off.

ATC integrates the aircraft into the air traffic flow under consideration of superior aspects of the *Air Traffic Management* (ATM).

Today activities for a take-off are initiated by a pilot (see fig. 6). He requests a *Start-Up* from ATC. Expecting that the aircraft will need a certain time to reach the runway and under consideration of ATM aspects, permission for *Start-Up* is given. Then the pilot requests the permission for *Push-Back* operation (and *Spool-Up* of the engines) from the Apron Control. The apron controller tries both to hand-over the aircraft in due time to the Tower Control and to coordinate the push-back and the taxi operation of this aircraft with all activities necessary for other aircraft. Under this consideration push-back and taxi instructions are given to the pilot. After hand-over suitable instructions for taxiing, line-up, and take-off have to be given from the PL to meet all constraints mentioned above.

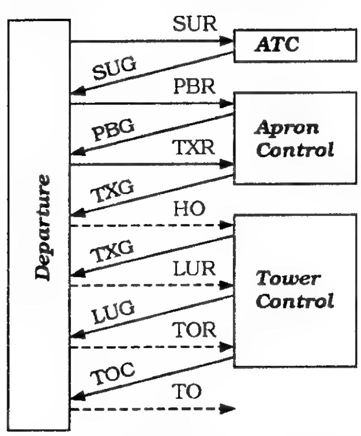


Fig. 6. Present operational procedures from start-up request to take-off

Dotted lines: observed events.

Abbreviations: SUR = Start-Up Request, SUG = Start-Up Given, PBR = Push-Back Request, PBG = Push-Back Given, TXR = Taxi Request; TXG = Taxi (Clearance) Given, HO = Hand-Over; LUR = Line-Up Readiness; LUG = Line-Up Given; TOR = Take-Off Readiness; TOC = Take-Off Clearance; TO = Take-Off

The present operational procedures (OP) are characterized by the fact that there is only a weak co-ordination between the institutions. Whether a certain slot can be met or not, is often noticed not before the aircraft has started its taxi-operation. Thus the sequence of departures reaching the runway is not appropriate to all constraints and to the arrivals landing in the near future because it results from the activities of AC driven by pilot requests.

4.1.3 Given and Generated Information

Human planning as well as Automatic Planning has to be based on relevant information that is obtainable (measurable, observable) from the system for which planning should be done, called *world*. In order to reduce the quantity of information to a manageable amount, a certain level of abstraction is used which describes the *state of the world* through the sets of *Events* (***E***), of known arrivals ***A*** and departures ***D***, *Constraints* (***C***), and the *previous Plan* (***P***).

An *event* $e \in E$ can be:

- ☐ a departing aircraft starting a new step of its OP (SUR, SUG, PBR, ..., TO; see fig. 6)
- ☐ the announcement of a new (up to the present time t_0 unknown) aircraft
- ☐ a changed prediction for the touch-down time of an arrival
- ☐ the actual touch-down time of an arrival
- ☐ an alteration of other time constraints for the present or the future

Any planning has to take into account the following constraints:

- ☐ Wake vortex separations between two arriving or departing aircraft in dependence on their weight classes; stored in form of matrices W.
- ☐ SID separations between all navigation aids (see fig. 7). The separation between two departing aircraft R_{DD} is than the minimum separation of the common flight path.
- ☐ Departures slots S_D
- ☐ Minimum runway allocation times for the known arrivals Y_a and departures Y_d
- ☐ Minimum sum times for all future steps of the OP O_D for every departure from D.
- ☐ Some constraints resulting from the interdependence between the runway 18 and the runway system 25/07 that will not be explained in detail.

For the computation of O_D a model M of the OP is needed, which is a set of tables containing average times for steps of the OP (e.g. push-back and taxi times) distinguished both for different areas or gates of the apron and aircraft weight classes (heavy, medium, light) or types (B747, A320, etc.). Since some constraints result from the *events* they change with time.

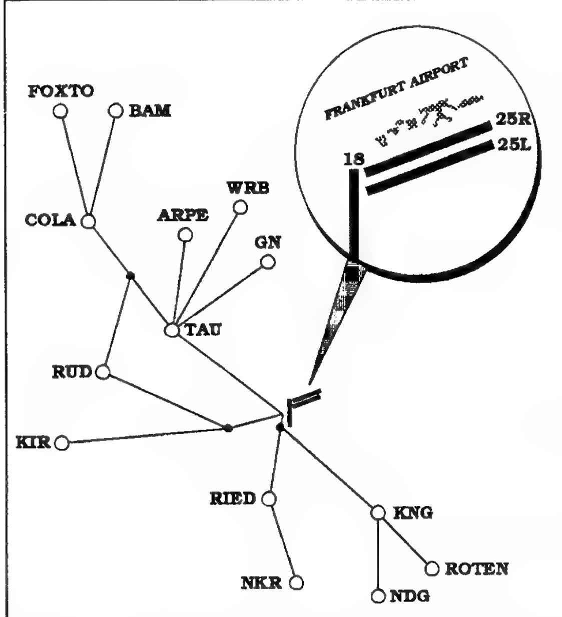


Fig. 7. (Simplified) Structure of the Standard Instrument Departure Routes from 25L+25R of the Frankfurt Airport

Navigation aids: FOXTO, RAM, WRB, ...,NDG, e.g. WRB=Warburg

It should be remarked that there are two kinds of constraints: the "strong" constraints that are safety critical and must not be violated in any case (wake vortex separations, minimum runway allocation times), and the "weak" ones that can be violated "slightly", for instance in cases where no other solutions exist.

When considering plans it is useful to distinguish between primary plans ***P*** for runway allocation and derivative plans ***p***=*p*(***M***,***P***) which correspond to the departures from ***D***. Any ***p*** contains latest times for the SUG, PBG, TXG, HO, and LUR of the OP of the respective aircraft.

4.1.4 Planning Problems

Inherent Complexity

Especially in domains where time stressed decisions are required the complexity of a planning problem is very important.

If occupancy planning is considered as a schedule problem of one runway of n_D departing aircraft there are

$$N(n_A, n_D) = n_D! \binom{n_D + n_A}{n_A} = \frac{(n_D + n_A)!}{n_A!}$$

possible solutions, where n_A is the number of arrivals and n_D the number of departures known. For the parallel runway system N increases by 2^{n_D}. For our application $n_A \leq 10$ and $n_D \leq 10$ seem to be realistic figures, so that N is about $6.8 \cdot 10^{14}$ in worst case. Of course, usually it is not necessary to investigate all possible sequences since almost every constraint violation of an aircraft allows to cut the search tree.

Beyond that, sizes of the occupancy intervals have to be determined by $2n_D$ parameters.

Inherent Uncertainty

Some of the constraints which have to be considered by planning are based on future *events*. The time when a certain *event* will occur has to be predicted either by the planning system itself (by applying ***M*** on past *events* from ***E***) or by other external systems, e.g. the landing time prediction system. However, *event times* cannot be predicted accurately. Especially in those case where the occurrence of an *event* requires human decisions and sequences of operations, the accuracy is low.

More generally, the main reasons for uncertainty are:

- the inaccurate assessment of *world* signals
- the inaccurate modeling (***M***) of the world
- the incomplete knowledge about future *events* caused by the dynamic of the *world*

Required Similarity of Subsequent Plans

The prevailing uncertainty leads to the fact that the behavior of the planned *world* often differs from the real one. Expected *events* do not take place at the times which were predicted or unforeseen *events* occur. So a previous plan $\boldsymbol{P}(t_{-1})=\boldsymbol{P}_{-1}$ might not be suitable after a new *event* has happened at t_0 and a new plan $\boldsymbol{P}_0$ has to be made.

In order to realize a certain plan, a sequence of future operational actions is necessary which have to be done within certain intervals. In general, some of these are scheduled later than t_0. So a replanning at t_0 might change or add some future actions of t_0. Since these actions have to be prepared and have to be processed mentally by the controllers, the subsequent plans have to be "similar" that means that the sequence of (near) future actions should change as less as possible.

Optimization and Evaluation Criteria

For the basic planning algorithms an optimization criterion is needed

- to eliminate undue solutions that violate strong constraints
- to permit slight violations of weak constraints
- to evaluate similarity between subsequent plans
- to consider prevailing uncertainty

The optimization criterion determines the difficulty and the complexity of the problem, especially whether the planning problem can be reduced to a discrete problem. For instance, if the optimization criterion is defined by

$$f_1(C(t_0)) = \sum_{i=1}^{n_D} t_{to,i}(C(t_0))$$

where $t_{to,i}(C(t_0))$ is the departure time of aircraft i, only $n_D!$ possible solutions have to be investigated;

whereas the function

$$f_2(C(t_0)) = \sum_{i=1}^{n_A+n_D} d_{i-1,i}^2(C(t_0))$$

(to try to obtain an uniform runway usage), where $d_{i-1,i}$ is the distance between the minimum occupancy intervals of subsequent aircraft, can only be optimized in a continuous parameter space.

It should be noted that the departure delays that originally should be reduced cannot be a term of the optimization criterion since the departure push-back times result from $p=p(P, M)$. However, average departure delay is influenced by a chosen optimization criterion and also by many other factors of the planning algorithms. Therefore an evaluation criterion is needed, too, based on the average departure delay which allows a comparison between different algorithms, optimization criteria etc.

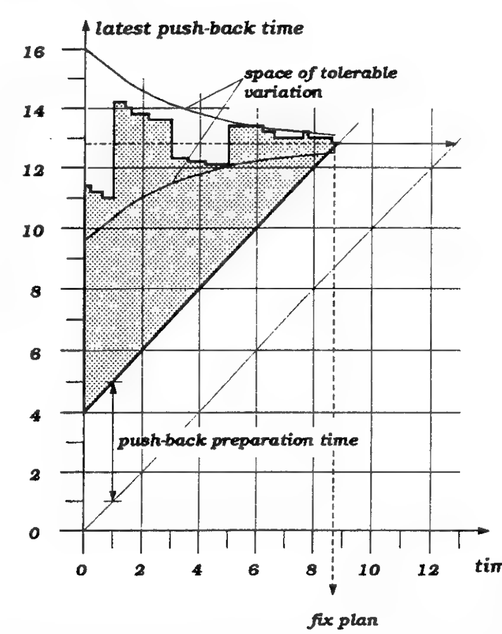


Fig. 8. Time Dependent Variation of the Calculated Latest Push-Back Time (According to $P(C(t))$) for a Certain Departure

4.2 An Approach to a Solution

4.2.1 Underlying Basic Principles

Reactive Planning

As explained in chapter 3 the future behavior of the *world* can only be predicted with uncertainty. The difference between the planned *world* and the real *world* is not continually observable but only at times at which *events* occur. Whether a certain *event* requires a "renewed" planning, has to be detected by a plan monitoring function. In this sense planning is "*event*-driven" (by *initial events*) and not subject to a certain time cycle.

The *initial events* for a new planning are

- ☐ announcement of new aircraft
- ☐ changed predictions for the occupancies of the arrivals
- ☐ each present or announced future alteration of time constraints
- ☐ severe delays (so that aircraft cannot meet their occupancies) of push-back or taxi operations

With every new planning process the *planning horizon* slides farther into the future. Therefore this principle was called *Event-Driven Recurrent Planning with Sliding Horizon.*

As a result of the consecutive planing cycles the planned latest push-back times vary significantly, however the ammount of the tolerable variation decreases with the progress of planning to an acceptable margin (fig. 8).

Planning in Consideration of Uncertainty

In order to avoid that plans often are getting inappropriate, even when the actual behavior of the world differs only slightly from the planned one, planning should be done in consideration of the prevailing uncertainty. This is possible by

- ☐ assuming that an event (especially the landings) will happen not at a certain time but within a certain interval T_{td} according to the known precision of the prediction. Thus touch-down of an arrival could be predicted so that the probability that the actual touch-down time t_{td} is within the predicted interval T_{td} is greater than a given threshold P_ϵ, i.e. $P(t_{td} \epsilon T_{td}) \geq P_\epsilon$. It is important that there is a general trend that renewed predictions for the same arrival produce T_{td}'s which are smaller than the previous ones.
- ☐ adding some heuristic values to the strong constraints (separations, minimum times for runway usage)
- ☐ modifying the optimization criteria
- ☐ designing a "pessimistic" function $p(P,M)$ assuming departures will need longer than the average push-back and taxi times[1)]
- ☐ expanding the minimum occupancy intervals for departures

4.2.2 Functional Structure of the Planning System

Superior Runway Allocation

Considering the specific runway configuration of the Frankfurt Airport the planning system was subdivided

[1)] this produces planned average departure delays

into a Coordinator Unit and two planning units: for the runway 18 and for the parallel runway system (fig. 7). The coordinator allocates the departures to a certain unit using a set of allocation rules. Beyond that it has to do some other tasks in connection with the planning algorithms of each unit (see following text).

Functional Structure of a Planning Unit

Each planning unit calculates a primary plan $\boldsymbol{P}$ for a set of departures $\boldsymbol{D}$ containing all departures which were allocated to the corresponding runway. This process is triggered by the coordinator after the occurrence of an initial event at time t_0. So planning has to consider the actual constraints $C(t_0)$. However, there is no guarantee that under these constraints a permitted plan really exists. As it is necessary for ATM to modify a certain constraint, like a departure slot, immediately in such a case, planning is done by two subunits:

- ☐ Sequencer: determination of permitted departure sequences
- ☐ Optimization Unit: optimization of occupancy intervals

Sequencer

The sequencer determines such sequences s_j of aircraft for which permitted occupancy intervals for the departures exist.

The allocation of a corresponding set of occupancy intervals (a preliminary plan $\boldsymbol{P}_p(s_j,\boldsymbol{R})$) is done by using a set of rules $\boldsymbol{R}$. When applying $\boldsymbol{R}$, it has not only to be guaranteed that a $\boldsymbol{P}_p(s_j,\boldsymbol{R})$ will be found for any s_j but, since in case that decisions are urgently required, the plans p for certain departures have to be derived immediately from the $\boldsymbol{P}_p$, the above mentioned aspects of plan stability and uncertainty have to be considered.

As in general a large set of permitted sequences exist and also in order to cut the search tree, an evaluation criterion $f(\mathbf{s})$ is used to select an subset of "sensible" sequences.

Optimization Unit

The Optimization Unit does not only consider whether a plan is permitted, but moreover other aspects such as:

- ☐ similarity to the previous plan
- ☐ robustness against uncertainty
- ☐ priority of aircraft
- ☐ violation of (weak) constraints.

Therefore an optimization function

$$g(C(t), \mathbf{v}) = \sum_i \alpha_i g_i(C(t), \mathbf{v}), \quad \sum_i \alpha_i = 1$$

is used where each g_i measures a certain aspect depending on the $2n_D$ boundaries of the occupancy intervals that are stored in $\mathbf{v}$, and α_i is a weight factor.

Since in general there are a lot of (local) optimal solutions which minimize g, an advanced optimization technique is used which is able to find the global (or a sufficient) optimum. This can be Genetic Algorithms /23/ which were already used for some planning tasks /24/.

Genetic Algorithms are a metaphoric abstraction of natural biological evolution including natural selection, and mutation which are applied on individuals belonging to a population. Each individual l is a specific realization of occupancy intervals $\mathbf{v}_l$. The evaluation of $g(\boldsymbol{C},\mathbf{v}_l)$ controls the probability to produce a "child" which is mostly a small random variation of its "parent" except a mutation is simulated, which can be interpreted as a turn of a certain aircraft from one runway (of the parallel runway system) to the other or as a change of the aircraft sequence.

Every optimization process starts with the population of the previous planning, but with a changed evaluation of all individuals. During the planning process new individuals representing permitted solutions and generated by the sequencer are added consecutively to the population. At any time the best solution found so far is accessible from the database. The coordinator decides by consideration of the available push-back preparation times whether for a departure j the plan $p_j \epsilon \boldsymbol{p}$ has to be fixed (fig. 9). Then the optimization is continued with a set start time of the occupancy interval j.

Coordinator Tasks

Beside the allocation task the Coordinator Unit controls not only the planing processes in the planning units but also serves as an interface to the *world* (via other ATC systems) as well as to the controller displays. The Coordinator Unit

- ☐ updates the *event list* $\boldsymbol{E}$ and constraints $\boldsymbol{C(E)}$
- ☐ proves whether a new *event* requires a new planning
- ☐ derives the plans $\boldsymbol{p}$ from $\boldsymbol{P}$
- ☐ initiates the modifications of constraints in case no permitted $\boldsymbol{P}$ exists
- ☐ establishes priorities to those aircraft of which push-back is in preparation or already done, to avoid delays caused by changed plans

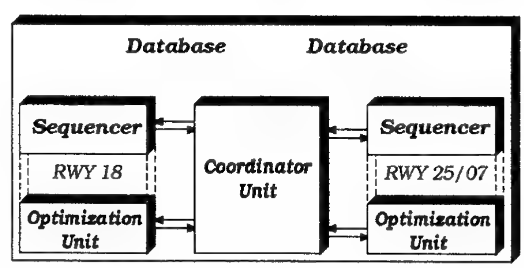


Fig. 9. Functional Structure of the Occupancy Planning System

4.2.3 An Example for Occupancy Planning

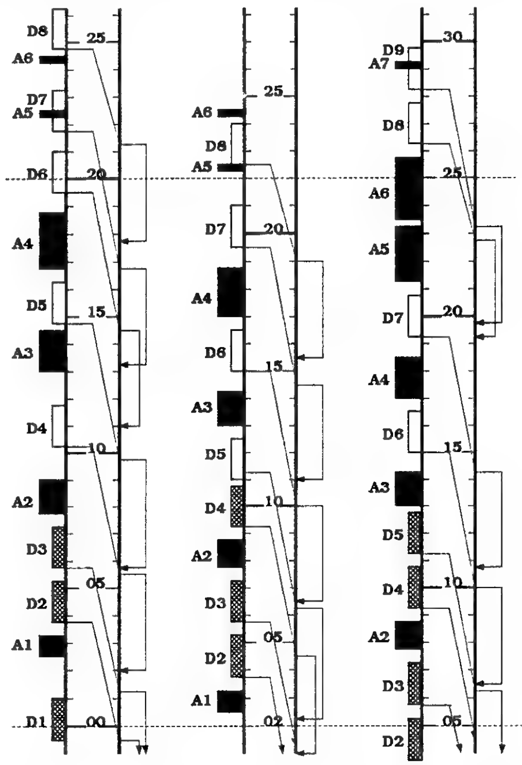
As explained in the previous chapters, the dynamic of an event-driven reactive planning leads to time-variable primary and derived plans ($\boldsymbol{P}$, $\boldsymbol{p}$). This can be demonstrated with an example for occupancy planning

at different times (fig. 10). In order to focus the attention on the essential points, the example was simplified in many ways. So only a single runway is considered and most of the constraints are not listed.

Figure 10 shows the occupancy planning at the times "00", "02", and "05" (minutes of an hour). The lower horizontal line marks the specific present time, the upper line the arrival prediction horizon. The arrivals above this line are unknown for the planning system. At time "00" occupancy planning only considers the arrivals A1, ..., A4 (left time line). The derived plans p of the gray marked departures D1, D2, D3 are fixed, because these aircraft have already started their push-back or their push-back preparation. Planning results for time "02" are shown in the middle. As a result of the decreasing prediction uncertainty D5 is now scheduled before A3. So the other departures are planned earlier, too. At time "05" A5, A6 and D9 are known to the planning system. This has the effect that D8 has to be planned later.

So the example shows that even if the departure sequence does not change, the planned occupancy intervals as well as derived latest push-back times alter from planning to planning, e.g. the values for latest push-back time for D8 change from "17:45", "15:30", to "19:15" (see also fig. 8)

Fig. 10. An Example for Occupancy Planning at Different Times



At the left side of every time line the runway occupancy intervals are disposed. The gray and white intervals symbolize the planned occupancies by departures D1, ..., D9 (the plan P), the black ones the predicted occupancies by arrivals A1, ..., A7. At the right side of the time lines the latest hand-over time and latest push-back time for each departure can be seen (elements of the plan p), e.g. at "00" for D5 these times are "09:45", "05:45" respectively.

4.2.4 An Approach to Distributed Planning

In section 4.1.4 it was mentioned that complexity of the scheduling problem increases by 2^{nb} for the parallel runway system. In order to reduce the necessary calculation time for permitted sequences, currently an approach for distributed planning is investigated which allows to plan in parallel and to consider only a part of the departure set.

According to this approach the runway allocations of the individual departures are fixed successivly[1)] and during the further planning process will not be changed any more[2)].

During a planning cycle both sequencers compete for a not allocated departure by adding (on trial) it to their departure sets (all aircraft which were allocated so far to this planner) and by searching for an optimal sequence. This is done under consideration of all constraints caused by all known arrivals, the departures from their departure sets, and the previous planned departure occupancies of the neighboring runway. Then the minimum value of both optimization criteria and the corresponding runway number is tentatively assigned to the aircraft. After evaluation of all departures in this way, the departure with the minimum value is selected. Its runway allocation (however not its occupancy interval) is fixed. That means, it is now actually added to a certain departure set.

This planning cycle is repeated as long until all departures are assigned to a certain runway. After termination both partial sequences are merged and the resulting sequence of occupancy intervals is delivered to the Optimization Unit via the database.

4.2.5 System Development and Evaluation

As explained in previous chapters occupancy planning requires a complex interaction of several subsystems. The influence of the implemented algorithms (sequencing, optimization etc.), of the several parameters and of the optimization criteria on the departure delays, resulting from the plans of an event-driven recurrent planning, is not obvious and there is no theoretical method known to quantify this influence in advance. Moreover, the evaluation of a certain realization of the planning system cannot be restricted to average departure delays because the frequency and the amount of violations of weak constraints, or the number of

1) if a departure has already started its push-back a once settled runway allocation will not be changed in future

2) Unfortunately, there is neither a guarantee to get the optimal solution nor, if there are only a few permitted solutions, to obtain any one in this way.

slots which might have to be rearranged, are important aspects, too.

Besides these technical aspects, the operational side has to be evaluated. So questions like, e.g.: what information should be displayed at which time (relating to a certain event time) in which way, have to be answered. This should be done as early as possible because this does not only has an influence on controllers' workload but a backward effect on the design of the planning algorithms.

The interdependence between planning algorithms and working procedures of the controllers can be investigated in the best way, by using a simulation system that is integrable into a realistic simulation environment including a tower mock-up which is already available at DLR. The simulation system which is currently under development generates all *events* based on realistic scenarios of arrival traffic and departure flight plans.

The workload of the controllers that is very important for their acceptance is moreover affected by plan stability. In order to decide whether (a certain realization of) the occupancy planning is able to generate "stable" plans a time-variable space of tolerable variation will be defined (fig. 8).

5 Conclusions

The management and control of complex systems by human operators can be improved considerably if they are assisted by an automated planner that considers all available information about future constraints, goals, events etc. as well as the present state of the system. Since in general the behavior of the *world* is not predictable with arbitrary accuracy a dynamic planning has to be designed that is characterized by information feedback and a permanently working situation assessment.

Depending on the characteristics of a given domain, influencing factors like e.g. : planning with uncertainty, situation assessment in real-time, planning with time-intervals and recurrent planning with sliding horizon have to be considered in the design and development of intelligent planning systems. If furthermore human operators have to interact with the planning support system, which is typical for the ATM domain, special provisions have to be designed from the beginning into the planning algorithm.

References

1) D. J. Musliner; J. A. Hendler; A. K. Agrawala; E. H. Durfee; J. K. Strosnider; C. J. Paul: The Challenges of Real-Time AI. COMPUTER (ISSN 0018-9162), Jan. 1995, Vol. 28, No. 1, pp. 58-66.
2) D. Böhme: Improved Airport Surface Traffic Management by Planning: Problems, Concepts, and a Solution – TARMAC. *H. Winter (Ed.): Advanced Technologies for Air Traffic Flow Management. DLR-Seminar Series, Bonn, Germany, April 1994.* in Lecture Notes in Control and Information Sciences, Springer Verlag, ISBN 3540198954
3) D. Böhme: Airport Capacity Enhancement by Planning of Optimal Runway Occupancies. Proceedings of the "39. Internationales Wissenschaftliches Kolloquium", Technische Universität Ilmenau (Thür.), Bd. 3, S.250-256, 1994, ISSN 0943-7207.
4) AGARD (Advisory Group for Aerospace Research & Development): Knowledge-Based Guidance and Control Functions. AGARD Advisory Report No. 325, January 1995, ISBN 92-836-1009-1
5) F. M. Brown: The Frame Problem in Artificial Intelligence. *Proc. of the 1987 Workshop, Lawrence, KA. Los Altos, CA (Morgan Kaufmann) 1987.*
6) L. Kaelbling: An architecture for intelligent reactive systems. *Proceedings 1986 Workshop Reasoning about Action,* 1986.
7) B. Abramson: A Decision -Theoretic Framework for Integrating Sensors into AI Plans. *IEEE Trans. on Systems, Man, and Cybernetics,* Vol. 23, No. 2, March/April 1993, pp 366-373.
8) E. Werner: Planning and Uncertainty: On the Logic of Multi-Agent Plans and their Entropy. *W. Brauer, D. Hernández (Eds.): Verteilte Künstliche Intelligenz und kooperatives Arbeiten. 4. Internationaler GI-Kongreß Wissensbasierte Systeme, München, 23.-24. Oktober 1991, Proceedings,* Informatik Fachberichte 291, Springer-Verlag 1991, pp 77-89.
9) D. Dippe: TARMAC – Part of a new Airport Surface Traffic Control System. Airshow Canada Symposium, Vancouver, August 1991
10) J. F. Allen: Towards a General Theory of Action and Time. *Artificial Intelligence 23,* 1984, pp 123-154.
11) L. A. Zadeh: Outline of a Computional Approach to Meaning and Knowledge Representation Based on the Concept of a Generalized Assignment Statement. *H. Winter (Ed.): Artificial Intelligence and Man-Machine Systems. Proc. of an International Seminar Organized by Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt (DFVLR). Bonn, Germany, May 1986.* in M. Thoma, A. Wyner (Eds.): Lecture Notes in Control and Information Sciences, DFVLR 80, Springer-Verlag, 1986, pp 198-211.
12) N. V. Findler, R. Lo: An Examination of Distributed Planning in the World of Air Traffic Control. *Journal of Parallel and Distributed Computing,* Vol. 3, pp 411-431, 1986.
13) Gerling, W.: Conflict Recognition in Air Traffic Control. Proceedings of Scientific Seminar of the Institut of Flight Guidance, DLR "From Sensors to Situation Assessment", Braunschweig, May 3rd-4th, 1995.
14) K. Inderfurth: Nervousness in inventory control: analytical results. OR Spektrum Vol. 16, 2(1994), pp.113-123, Springer 1994.
15) R. Rymon, B. L. Webber, J. R. Clarke: Progressive Horizon Planning-Planning-Exploratory-Corrective Behavior. *IEEE Trans. on Systems, Man, and Cyb., Vol. 23, No. 6, pp. 1551-1560, Jan. 1994*
16) K. Platz, U. Brokof: Optimising Air Traffic Flow at Airports. *H. Winter (Ed.): Advanced Technologies for Air Traffic Flow Management. DLR-Seminar Series,*

Bonn, Germany, April 1994. in Lecture Notes in Control and Information Sciences, Springer Verlag, ISBN 3540198954

17) F. v. Martial: Activity Coordination via Multiagent and Distributed Planning. *W. Brauer, D. Hernández (Eds.): Verteilte Künstliche Intelligenz und kooperatives Arbeiten. 4. Internationaler GI-Kongreß Wissensbasierte Systeme, München, 23.-24. Oktober 1991, Proceedings,* Informatik Fachberichte 291, Springer-Verlag 1991, pp 90–101.

18) E. D. Sacerdoti: Planning in a Hierarchy of Abstraction Spaces. *Artificial Intelligence Journal,* Vol. 5, 1974, pp 115–135.

19) D. E. Wilkins: Hierarchical Planning: Definition and Implementation. *Proc. ECAI-86,* pp 466–478, 1986.

20) E. H. Durfee, V. R. Lesser: Incremental Planning to Control a Blackboard Based Problem Solver. *Proc. of AAAI,* Pittsburg, PA, August 1986, pp 58–64.

21) H. Winter: Artificial Intelligence in Man-Machine Systems and Automation. *H. Winter (Ed.): Artificial Intelligence and Man-Machine Systems. Proc. of an International Seminar Organized by Deutsche Forschungs- und Versuchsanstalt für Luft- und Raumfahrt (DFVLR). Bonn, Germany, May 1986.* in M. Thoma, A. Wyner (Eds.): Lecture Notes in Control and Information Sciences, DFVLR 80, Springer-Verlag, 1986, pp 1–22.

22) A. R. Odoni: Issues in Air Traffic Flow Management. *H. Winter (Ed.): Advanced Technologies for Air Traffic Flow Management. DLR-Seminar Series, Bonn, Germany, April 1994.* in Lecture Notes in Control and Information Sciences, Springer Verlag, ISBN 3540198954

23) J. C. Potts, T. D. Giddens, S. B. Yadav: The Development and Evaluation of an Improved Genetic Algorithm Based on Migration and Artificial Selection. *IEEE Trans. on Systems, Man, and Cyb., Vol. 24, No. 1, pp. 73–86, Jan. 1994*

24) H. Kopfer, G. Pankratz, E. Erkens: Entwicklung eines hybriden Genetischen Algorithmus zur Tourenplanung. *OR Spektrum (1994) 16:21–31,* Springer-Verlag 1994.

25) U. Völckers: A Taxi and Ramp Management System (TARMAC) – Advanced Systems for Air Space Management. Canadian Conference on Electrical And Computer Engineering, Montreal, Sept. 17–20 1989.

BIBLIOGRAPHY

This Bibliography has been prepared by the Fachinformationszentrum der Bundeswehr (FIZBw), Dezernat 3 in cooperation with the Lecture Series Director, using the Fachinformations-Dokumentationssystems (FIDOS), to support the Lecture Series 200.

I. Document Number VC 0008: Conference Proceedings

"Advanced Computer Aids in the Planning and Execution of Air Warfare and Ground Strike Operations"

AGARD / Advisory Group for Aerospace Research and Development / Avionics Panel

1987, AGARD CP-404, Suppl

SUMMARY: The pace and complexity of modern air warfare are reaching the point where advanced computer aids are becoming essential to assist the aircrew in the aircraft and the commander on the ground in performing functions that hitherto had been considered to be their prerogative. Computers are already extensively in the operation and control of specific types of equipment, such as advanced weapon systems, radars, electronic warfare and communications systems. in the broader context there are still many areas which rely heavily on human decision making and where the use of computers will have considerable impact in the future. The increasing use of AI techniques, including IKBS and Exp Syst will at one extreme allow decision making to be increasingly automated or controlled by non-expert personnel, and at the other extreme greatly extend the capabilities of military commanders by presenting information in a timely manner and by making rapid assessment of alternative strategies. The successful application of computers should provide improved effectiveness, flexibility and reliability of men and equipment resulting in a saving of resources and personnel.

II. Document Number FA 4851: Conference Proceedings

"Knowledge Based System Applications for Guidance and Control"

AGARD / Advisory Group for Aerospace Research and Development / Guidance and Control Panel

1991, AGARD CP-474

SUMMARY: The combination of increasing military system and task complexity, in the face of inherent human limitations has set the stage for development of innovative system integration approaches involving the use of knowledge based technology. The field of Artificial Intelligence (AI) is becoming solidified as a science and the technological aspects are developing rapidly. The implications for guidance and control are enormous. Practical applications of AI are critically dependent on advanced architectures, computer processing techniques and integration concepts. Recent advances in digital computation techniques including data base management. represent the core enabling technology necessary for the development of highly innovative design concepts, which will ultimately lead to major new military capabilities. Efficient tactical information management and effective pilot interaction are essential. Pilot decision aiding, combat automation, sensor fusion and on-board tactical battle management concepts offer the opportunity for substantial mission effectiveness improvements. Although real-time tactical military applications are relatively few in number, several exploratory and advanced development efforts are underway. Practical military applications of AI technology are of primary interest. Projected military capability enhancements along with AI limitations were considered. Operational implications, and critical design trade-offs were also emphasized. This symposium provided a timely forum for assessing the overall state-of-the-art of AI applications in the guidance and control area. A round table discussion to identify application issues and opportunities was held.

RELEVANT PAPERS IN THIS REPORT:

1.) Retelle, John, P jun; Holmes, Douglas, I: "The Pilots Associate/Exploiting the Intelligent Advantage"

> The Pilot's Associate program has provided a series of technology demonstrations of the potential of integrating intelligent systems and artificial intelligence technology into modern avionics systems. The Defense Advanced Research Projects Agency and the United States Air Force have provided funding

and program management to determine the potential increases in mission effectiveness from such a system. The Pilot's Associate effort pursued by Lockheed and its partners has produced not only prototypes for advanced systems, but also new insights into the nature of the systems themselves as well as new approaches for quickly producing software for these systems. The rapid prototyping methods that have been utilized have also provided the ultimate consumers - the pilots - with significant awareness of the operation of the Pilot's Associate, and with many opportunities to improve the requirements for such a system. This paper describes the evolution of Lockheed's Pilot's Associate System approach leading to the current system configuration. Also described are some lessons learned from managing a large software development team assembled to produce an unprecedented system.

2.) Luise, Federica; Dabbene, Danilo: "Path Generation and Evaluation for a Mission Planning Expert System"

The aim of this paper is to describe the experience with the problem of path generation and evaluation for a multitarget air to ground planning system working in a limited geographic scenario. The generation of a flight plan for a ground attack mission in visual flight is a process that finds a path joining a sequence of way points which connect the take off field with the landing one through the planned targets of mission. A set of constraints must be taken in account to build the paths that will become flight plans. These constraints will be used in the generations of plans together with a set of criteria for evaluating the quality of the plans itself.

3.) Onken, R: "Konwledge-Based Cockpit Assistant for IFR Operations"

A knowledge-based cockpit assistant for IFR (Instrument Flight Rules) operation is presented, aimed at improvement of situation assessment and performance increase by computer aids for flight planning and plan execution. Here, situation assessment also includes monitoring of the pilot's own activities. The modular system structure is described as well as the individual system modules. The cockpit assistant was tested in a flight simulator by professional pilots under realistic IFR-scenarios. The concept of the test design as well as test results are presented. The system design goals are mainly confirmed by these results.

4.) Teegen, Uwe: "Constraint Management Requirements for online Aircraft Route Planning"

In the future, the cooperation of pilot and controller will change. Technical advances contributing to this change are a more intelligent airborne Flight Management System (FMS) and a datalink connecting the FMS and Air Traffic Control (ATC) . Against this background, concepts for a Experimental Flight Management System and its humancentered system design approach are described. Combined with a basic scenario of future Air Traffic Management (ATM) the requirements for an airborne constraint management subsystem are developed. The fundamentals of aircraft route planning and system operation including considerations on the interaction between constraint management and man-machine-interface are discussed and an on-line algorithm for aircraft route planning is presented. The paper

also describes the present state of a software prototype and the software and hardware employed.

5.) Adams, Milton, B; Beaten, Robert, M: "Planning and Planning Management for Autonomous and Semi-Autonomous Vehicles"

6.) Wilber, George, F: "Intelligent Real-Time Knowledge Based Inflight Mission Management"

This paper describes the problems and issues of developing a tactical mission manager. It discusses development aspects of intelligent real-time avionics, and outlines an efficient real-time AI (Artificial Intelligence) methodology and implementation for the development of the intelligent systems. It also outlines advanced software development techniques and provides an overview of related Boeing research efforts.

7.) Belkin, Brenda, L; Stengel, Robert, F: "Knowledge Acquisition for Expert Systems using Statistical Methods"

8.) Perez, Manuel; Gemoets, Legoldo; MacIntyre, Robert, G: "Knowledge Extraction Methods for the Development of Expert Systems"

The development of expert systems require the use of engineering techniques which can be used to efficiently and correctly extract the domain knowledge resident within the human expert. To apply these techniques, certain conditions must be met. These conditions are that the candidate expert system domain must be suitable for implementation, that there be a knowledge engineer with a certain level of domain knowledge, and that the right human domain experts be selected in the expert system development effort. This paper presents a semi-sequential approach to development of techniques which can be used to extract the knowledge from the human expert. Presented are both direct and indirect methods which a knowledge engineer can use to extract this knowledge.

9.) Cross, S, A; Grisoni, M: "A Methodology for Producing Validated Real-Time Expert Systems"

VORTEX (Validation Of Real Time Expert Systems) is an experimental methodology for building validated expert systems. It considers validation to be an exercise in building confidence in a system in the users, producers, experts and developers. It identifies the components of validation concerning each participant and techniques for achieving validation and embeds them in a life-cycle suitable for a novel technology. This paper presents the essential points of the methodology and some experiences from the airborne ASW (Anti Submarine Warfare) application developed in paralle with it.

10.) Trillas, E; Delgado, M; Verdegay, J, L; Vila, M, A: "A Review of some Aspects on Designing Fuzzy Controllers"

The paper focus both theoretical and practical aspects of the fuzzy control systems according to the following scope. First, foundations of the fuzzy controllers, and the different ways for implementing them, are described. Second, we concentrate ourselves in the management of the information, that is, the way in which the inferences are made from the expert's knowledge. Usually this is carried out by means of the Generalized Modus Ponens for which, the so called implication

function is the main tool to handle it. Hence, as depend on the selected type of implication one has a different version of inference, finally, the possible implications functions and the consequences from its use are analyzed and discussed.

11.) Golden, J, B; Whitehead, B, A: "A neural Network for the Analysis of Aircraft Test Data"

With the advent of the USAF's Advanced Tactical Fighter and NASA's National Aerospace Plane, demands for concise test data reduction and interpretation will increase beyond the capabilities of current methodologies. As mission complexity increases it becomes apparent that real time data analysis for flight safety, mission control and test conduct becomes a necessary tool. A neural network is a biologically inspired mathematical model, which can be represented by a directed graph, that has the ability to learn through training. Neural networks have many advantages over current aviation computing systems including the ability to learn and generalize from their environment. Neural networks are excellent for parameter estimation and recognizing patterns in signal data. This research discusses a prototype system designed and implemented at the University of Tennessee Space Institute to discover patterns in test data from an engine test cell in order to determine if any part of the system is in failure. The results of this research show that a neural network can be used for fault diagnosis in an engine test cell when the problem of fault monitoring and diagnosis is seen as one of pattern recognition. A two layer semilinéar feed-forward neural net is able to separate simulated sensor data into normal and abnormal classes and the addition of a hidden layer makes the network more resistant to noise and improves the ability of the network to classify the type of fault that is occuring.

12.) Corbin, M, J; Butler, G, F: "An Ada Framework for the Integration of KBS and Control System Simulations"

The application of Ada and an object-oriented approach to the design and construction of advanced defence systems are both attracting increasing attention. The Ada language contains some support for object-oriented programming but has some notable deficiencies. This paper shows how to overcome these deficiences by providing a library for object-oriented development in Ada (OODA) which contains facilities to create and manipulate objects and provides support for more general relationships between objects. One of the initial applications of this library has been to design a framework for integrating KBS with control system simulations comprising mixed continuous and discrete time elements. Using this framework it is possible to study the interactions between a Knowledge Based controller and the other more conventional elements of the closed loop to any level of detail required.

13.) Midollini, B; Torelli, P, L; Balzarotti, G: "Evaluation of the optimal Homing Point for Missile Guidance"

One of the main problems arising in the field of missile guidance is the automatic search and detection of targets, in order to gather the necessary information for the correct homing of the missile. This problem is commonly approached by mounting a seeker (usually an IR seeker) with an image and data

processor on board of the missile. The ground scene, as recorded by the seeker, will present isolated targets and target formation, together with a high level of clutter which produces a number of false alarms. Thus, it is necessary to provide the image and data processor of the missile with algorithms which can automatically eliminate the clutter and the false alarms and detect the true targets. Furthermore, in order to have an effective shot, only one formation among the various in the scene must be cued to the navigation and the weapon systems of the missile, the choice of it depending on the shape, the disposition and the distance between targets of the formation itself. This paper presents an algorithm developed to evaluate the optimal point for releasing the ammunition. The algorithm is further illustrated by applying it to a practical case and showing the results of this simulation.

14.) Huang, Chien, Y; Lodaya, Manikant, D: " Design and Simulation of an Advanced Airborne Early Warning System"

The results of the design and simulation of an advanced airborne early warning (AEW) system are presented. The approach is based on modeling operator's reasoning and decision processes as well as battlefield strategies. The tasks are divided into threat assessment and tactical planning. The implementation of these subsystems is carried out in a generic expert-system shell developed specifically for this purpose. The AEW crew is provided with an advanced display that monitors all transactions. The functionalities of this prototype AEW system are demonstrated using an advanced simulation facility. Simulation results show that decision automation can be accomplished in real time and can prove to be a valuable tool in an airborne early warning environment.

15.) Halski, Donald, J; Landy, Robert, J; Kocher, James, A: "Integrated Control and Avionics for Air Superiority/A Knowledge-Based Decision-Aiding System"

16.) Mackall, Dale, A; Allen, James, G: "A Knowledge-Based System Design and Information Tool for Aircraft Flight Control Systems"

17.) Koch, R; Bader, R; Hinding, W: "A Study of an Integrated Image and Inertial Sensor System"

The target approach over large distances of a cruise missile type vehicle, which is equipped with an imaging infrared sensor aided inertial navigation system, is examined in simulation. The study is based on the idea of using the same image sensor for navigation update and target recognition with subsequent tracking. Knowledge based methods are found to play a key role in solving the difficult image interpretation task for real world scenery. The final extraction of navigation data from the processed and interpreted IR-image information, and their combination with the inertial sensor data is based on conventional optimization and filtering techniques. The study shows that the combined information leads to an improvement of navigation data. Filtering techniques are found to be capable of quantitatively estimating major error sources inherent to the gyros and accelerometers.

III. Document Number FA 8801: Conference Proceedings

" Machine Intelligence for Aerospace Electronic Systems"

AGARD / Advisory Group for Aerospace Research and Development / Avionics Panel

1991, AGARD CP-499

SUMMARY: A large amount of research has been conducted to develop and apply Machine Intelligence (MI) technology to aerospace applications. Machine Intelligence research covers the technical areas under the headings of Artificial Intelligence, Expert Systems. Knowledge Representation. Neural Networks and Machine Learning. This list is not all inclusive. It has been suggested that this research will dramatically alter the design of aerospace electronics systems because MI technology enables automatic or semi-automatic operation and control. This symposium was organized to present the results of applying MI technology to aerospace electronics applications. The symposium focused on applications research and development to determine the types of MI paradigms which are best suited to the wide variety of aerospace electronics applications.

RELEVANT PAPERS IN THIS REPORT:

1.) Roberts, K: "TACAID (TACtical AID) , a Knowledge Based System for Tactical Decision making"

British Aerospace has long been aware of a possible future requirement to embed Artificial Intelligence (AI) technology in its conventional airborne systems, in order to achieve the complete mission system on future fighter aircraft. We have followed a development life-cycle of prototyping and evaluation to achieve these aims. The first of these prototypes has already been presented within this forum. The second prototype, TACAID, which is discussed in this document, uses an identical scenario to explore the coupling of high level heuristic reasoning with conventional modules that produce optimised steering and firing cues, and enemy and own kill probabilities against selected targets. Our aims were to assess the value of embedded AI. TACAID will support sensor fusion, situation assessment, tactical planning, sensor management, utilities management and pilot interface.

2.) Chapman, George, B; Johnson, Glenn; Burdick, Robert: "Automated Threat Response Recommendation in Environments of High Data Uncertainty Using the Countermeasure Association Technique (CMAT)"

This paper discusses the CounterMeasure Association Technique (CMAT) system developed for the Air Force, which is used to automatically recommend countermeasure and maneuver response to a pilot while he is under missile attack. The overall system is discussed, as well as several key technical components. These components include use of fuzzy sets to specify data uncertainty, use of mimic nets to train the CMAT algorithm to make the same resource optimization tradeoffs as made in a database of library of training scenarious, and use of several data compression techniques to store the countermeasure effectiveness database.

3.) Self, Arthur, G; Bourassa, Gregory: "Future ESM Systems and the Potential for Neural Processing"

The projected radar electromagnetic environments in the future

include: higher pulse densities, frequencies to 40 Gigahertz and higher, stable, jittered, staggered, and pseudo-random pulse repetition intervals with multiple frequencies, spread spectrum techniques, multiple agile radar beams and multi-mode missile seekers. Electronic Support Measures (ESM) concerns the passive detection and identification of radar signals. Thus, an ESM system which can measure such signal characteristics will most likely flood its main processor with information to such an extent that it may not be able to cope. A number of likely solutions exist ranging from special purpose hardware to new processing techniques. In this paper, a radically different processing approach is reviewed, namely that of neural network. This paper will indicate the likely applicability of a neural processing approach to a range of ESM functions together with results from some preliminary proof-of-concept investigations.

4.) Reibling, Lyle, A: "Neural Network Solutions to Mathematical Models of Parallel Search for Optimal Trajectory Generation"

A difficult problem in search applications is computing the optimal aircraft trajectory in real-time onboard a high performance aircraft, where the objective is to increase the aircraft survivability and mission effectiveness by penetrating enemy threats and minimizing threat radar exposure. Optimal trajectory generation is achieved by searching all possible paths in a multidimensional search space for the path with the smallest accumulated performance measure. This research has investigated computer architectures and algorithms characterized by massive parallelism which can solve trajectory generation problems. An artificial neural network is defined which computes solutions to field theory. Experimental investigation of this technique has shown promising results. This paper describes the problem and solution method in envisioning massively parallel architectures which incorporate mathematical physics models for trajectory generation applications.

5.) Daysh, Colin; Corbin, Malcolm; Butler, Geoff; Duke, Eugene, L; Belle, Steven, D; Brumbaugh, Randal, W: "A NASA-RAE Cooperation in the Development of a Real-Time Knowledge Based Autopilot"

As part of a US/UK cooperative aeronautical research programme, a joint activity between NASA Ames-Dryden and the Royal Aerospace Establishment on Knowledge Based Systems (KBS) has been established. This joint activity is concerned with tools and techniques for the implementation and validation of real-time KBS. This paper describes the proposed next stage of this research, in which some of the problems of implementing and validating a Knowledge-Based Autopilot (KBAP) for a generic high-performance aircraft will be investigated.

6.) Gustavson; Steven, C; Little, Gordon, R: "Locally Linear Neural Networks for Aerospace Navigation Systems"

Neural network software simulations for the representation and prediction of aircraft inertial navigation system (INS) data were developed. These simulations were evaluated using flight test data that sampled INS outputs at a standard rate for neural network testing and at half this rate for neural network training. The simulations used both locally linear neural

networks and backpropagation-trained neural networks. Locally linear neural networks have several desirable properties for this application, including interpolation of the training data and representation of linear relationships. For the flight test data two milliradian testing accuracy was generally achieved with five successive and prior INS heading, pitch, and roll increments as inputs.

7.) Piers, M, A; Donker, J, C: "A Knowledge-based Assistant for Diagnosis in Aircraft Maintenance"

As a result of the demands upon maintenance organisations to increase availability of aircraft and the increasing complexity of aircraft systems, a need for tools and facilities to enhance the effectivity and efficiency of maintenance emerges. The National Aerospace Laboratory NLR performs applied research with concern to the use of knowledge based systems for condition monitoring and diagnosis in complex technical systems. This paper describes a feasibility study (KADAM) on the application of knowledge based systems for diagnosis of complaints in aircraft systems. The specific application selected for the KADAM project (Knowledge-based Assistant for Diagnosis in Aircraft Maintenance) is a knowledge based system to be used by ground engineers for troubleshooting of an aircraft airconditioning system. This paper will address the approach taken in the project and review the results, including the design of the proof-of-concept system. Particular attention will be paid to the identification and formalisation of methods for diagnosis.

8.) Donker, J, C: "Reasoning with Uncertain Information and Incomplete Information in Aerospace Applications"

In many real-life application areas such as aerospace, decisions have to be taken based on imperfect knowledge. If decision makers are to be supported by computer systems, it is desirable that this type of knowledge can be represented. In the past years, new methods have been developed to represent various kinds of imperfectness, such as incompleteness, inexactness or uncertainty. Since 1987, NLR cooperates with the Theoretical Informatics Group of Delft Technical University to study ways in which methods to represent and reason with uncertain or incomplete information can be developed which are both theoretically well-founded and practically applicable. NLR efforts are directed towards problems expected in practical use of those methods. In 1990, we investigated the applicability of the Dempster-Shafer theory. We modeled parts of the initiation and identification problems in multi-radar tracking. The application will be described in this paper. It will be shown that the Dempster-Shafer theory promises improvements over the Bayesian approach, but also that the latter currently is a more advanced theory than the former. NLR = National Aerospace Laboratory.

IV. Document Number FA 8830: Lecture Series

" Artificial Neural Network Approaches in Guidance and Control"

AGARD / Advisory Group for Aerospace Research and Development / Guidance and Control Panel

1991, AGARD LS-179

SUMMARY: Ever increasing operational and technical requirements have to highly integrated flight, guidance and control, and delivery systems. The effective implementation of functions makes the fusion and interpretation of sensor and the multifunctional use of sensor information inevitable. Neural networks, consisting of parallel elements, hold great promise for guidance, navigation and control applications because of their ability to learn and acquire knowledge. The Lecture Series will bring together a group of NATO nation speakers with outstanding experience this new area of technology. First they will review the fundamentals of neural networks to serve as background so that advances in this new, rapidly evolving technological area be both understood and appreciated. They will then discuss number of related applications of direct benefit to the attendees. This Lecture Series, sponsored by the Guidance and Control Panel of AGARD, has been implemented by the Consultant and Exchange Programme.

RELEVANT PAPERS IN THIS REPORT:

1.) Simpson, Patrick, K: "Neural Network Paradigms"

> Building intelligent systems that can model human behavior has captured the attention of the world for years. So, it is not surprising that a technology inspired by the mind and brain such as neural networks has generated great interest. This chapter will provide an evolutionary introduction to neural networks by beginning with the key elements and terminology of neural networks and then developing the topologies, learning laws and recall dynamics from this infrastructure. The perspective taken in this paper is largely that of an engineer, emphasizing the application potential of neural networks and drawing comparisons with other techniques that have similar motivations. Mathematics will be relied upon in many of the discussions to make points as precise as possible.

2.) Bowen, B, Archie: "A Neural Network Design Methodology: Considerations and Issues for Design and Project Management"

> An artificial neural network (ANN) is a software implementation of a neural paradigm, and, therefore, such projects yield to many of the disciplines of software engineering. On the other hand, many issues that must be faced, as the project proceeds, are unique and require specialized knowledge to address. This paper is concerned mainly with the management of such projects, however in order to propose the management issues, it seems necessary to understand, at least superficially, the process of the design and implementation of a neural-based system. This paper therefore begins with a proposal for a methodology for the conduct of a project involving the choice, design, and implementation of a neural-based system. It outlines the issues that should be considered and resolved at each step of the project. Based on this methodology, a project management plan can be put in place. Such a plan calls for a set of milestones and design reviews for various levels of management (and the customer) and a corresponding document set designed to prove a milestone has been reached, and, finally, that the original requirements have been met.

3.) Gutschow, Todd; Hecht-Nielsen, Robert: "Processing Complexity of two Approaches to Object Detection and Recognition"

The computational complexity of a processing function is a driving factor in the implementation of that functions in an operational system. Artificial neural networks offer the potential for significant improvements in the computational complexity of a number of guidance and control functions. To illustrate such an improvement, this paper considers a comparison between two different approaches to object detection and recognition: a traditional approach employing a wide field of view and constant spatial resolution throughout the image sensing and processing chain, and a foveal approach utilizing a roving "eyeball" circularly symmetric sampling grid with a radially variant resolution in the processing chain. The rationale and characteristics of these two approaches are described and compared. Quantitative evaluations of the processing loads and data transfer rates are then carried out for both approaches. These processing requirements are then compared and the operational implications of this comparison are discussed. While this paper does not explicitly discuss the efficacy of the foveal approach, references to relevant research results in this regard are provided.

4.) Bowen, B, Archie: "Vision Systems for Guidance and Control/A Tutorial Overview"

5.) Wright, W, A: "Neural Networks for Military Robots"

This paper gives a short review of mobile robotic research and through the use of three case studies which describe, in brief, current research undertaken at three establishments, indicates the role that neural networks are playing in this process and hence the impact that they may have on the military environment. The three case studies are chosen to illustrate the advantage, in terms of speed, compactness, and adaptibility, of the use of these systems in the three essential functional areas for mobile robot control. These areas are localisation, path planning and obstacle avoidance. Although it is not intended, by presenting these case studies, to portray them as the extent of the state of the art in this field it is, however, hoped that they will give a clear idea of how and why neural networks are being used in this area.

6.) Simpson, Patrick, K: "Multisensor Data Fusion as applied to Guidance and Control"

Multisensor data fusion (MDF) is the synergistic application of data from several sources, typically sensors, toward a specific task. In the area of guidance and control data fusion plays a very important role. By combining the information from several sensors it is possible to improve the performance of guidance and control systems. Neural networks are ideally suited to applications where only a few decisions are required from a massive amount of data. In this sense, neural networks should play a crucial role in future data fusion systems. This paper will describe several methods of applying neural networks to data fusion, including: self-organizing hierarchical neural systems, multi-layer error correction learning networks, and single layer pattern completion systems. Application case studies will be examined to determine how researchers have

applied neural networks to data fusion. In addition, a discussion of feature representation and feature weighting will be provided.

7.) Gutschow, Todd; Hecht-Nielsen, Robert: "Advance Neural Network Architectures for Guidance and Control"

Several advanced neural network architectures are expected to be of significant value in guidance and control. This paper reviews three advanced neural network architectures (the graded learning network, the recurrent backpropagation network, and the hierarchical matched filter network) und briefly discusses how they might be applied to problems in guidance and control.

V. Document Number FC 0355: Conference Proceedings

"Combat Automation for Airborne Weapon Systems: Man/Machine Interface Trends and Technologies"

AGARD / Advisory Group for Aerospace Research and Development / Flight Mechanics Panel and Guidance and Control Panel

1993, AGARD CP-520

SUMMARY: Recent advances in combat automation technologies offer significant potential for improving overall mission effectiveness. Development of advanced situational awareness display concepts, parallel distributed computer architecture and tactival information fusion techniques have paved the way for new operational capabilities and weapon system employment tactics. Harnessing these innovative technologies is critically dependent upon establishing an effective and intuitive pilot vehicle interface.

Presentation of accurate situational data at the right time in an appropriate format remains a significant challenge. Effective combat systems must employ anticipatory control laws, data management and display techniques. Consequential trend information based on both current decisions and alternative courses of action is essential. A well integrated system must reconcile multiple and potential conflicting data sources relative to the current and projected tactical situation and aircraft state. Future manned fighter systems must be capable of providing automated command guidance and signal limiting when appropriate, e.g. ground collision avoidance cues, AOA/g limiting, etc. Additionally, future systems must also correctly harmonize the automatic functions consistent with the pilot's intention and total tactical situation.

It was decided by both the Flight Mechanics Panel and Guidance and Control Panel of AGARD that a jointly sponsored Symposium on these topics would be both timely and effective.

The Symposium addressed changing and possible future operational scenarios, advanced technology concepts, application issues and experimental development efforts and included sessions on: combat mission application, tactical decision aiding and information fusion, situation awareness, human capabilities and limitations, and design and evaluation of integrated systems. It closed with a Round Table Discussion on the prospects and limitations for combat automation.

RELEVANT PAPERS IN THIS REPORT:

1.) Gray, I, D: "Planning for Air to Air Combat"

Air combat planning has always proven very difficult because of the dynamic environment, intelligent adversaries, group operations and the incomplete nature of any information. Two approaches, those of 'expert systems' and classical adversary search are presented and compared. Searching is then described

and developed in detail. The implications of such an approach are considered for the future of air combat.

2.) Buffett, A, R; Wimbush, R, M: "Pilot Decision Aiding for Weapon Delivery/A Novel Approach to Fire Control Cueing Using Parallel Computing"

This paper describes the application of advanced technology, both hardware and software, to provide improved pilot Manmachine Interface (MMI) automation for the central function of an airborne weapon system, namely weapon release. The paper gives an overview of the need for automation/decision aiding in air-to-air missile fire control, by illustrating the way in which missile performance can vary greatly with the changes of engagement parameters which occur rapidly in an air-to-air combat scenario. Current methods of generating and displaying fire control cueing information to the pilot are described. A novel future approach, the use of an on-board missile fly-out simulation is presented. This relies upon the development of a simple, but sufficiently accurate, missile fly-out model, and the use of parallel processing to achieve the required 'faster-than-real-time' operation and multiple simultaneous cueing. The development of such a model, and its potential to provide an efficient and intuitive MMI for fire control cueing for future missiles and combat scenarios, is described.

3.) Pipe, Harvey, J: "A New Class of Mission Support for Combat Air-Crew"

In the next century, combat aircraft will be even more complex than those planned as current replacements; this is to counter increasingly competent aggressors, who may operate anywhere in the world. To tackle the need for a new class of mission support, UK Industry and the Ministry of Defence set up the Mission Management Aid (MMA) Project. By rapid prototyping of software, the functional requirements of the MMA, and also the real-time symbiosis between man and intelligent machine are being investigated. This paper covers the integration of an MMA into future combat aircraft, its operation, the core topics of Sensor Fusion, Situation Assessment (including Dynamic Threat Assessment) , Planning and Tactical Routing (with Defence/Attack Options Management) . Evaluation of the MMA is showing that better situation awareness is obtained, increasing mission effectiveness and survivability, and that overall the MMA is a vital integral system for future aviation.

4.) Wittig, T; Onken, R: "Pilot Intent and Error Recognition as Part of a Knowledge Based Cockpit Assistant"

A Pilot Intent and Error Recognition module as part of a knowledge based Cockpit Assistant System is presented, which is being developed at the University of the Armed Forces in Munich in cooperation with the Dornier company and implemented in a flight simulator. The system mainly supports the pilot crew with regard to the monitoring and planning task and provides assistance for a number of plan execution functions for the civil flight operation under Instrument Flight Rules. In this paper a short survey is given of the concept and the function of the Cockpit Assistant System. After that the structure of the Pilot Intent and Error Recognition will be described in detail. At the end, the integration of this module into the Cockpit Assistent System and the evaluation in a flight

simulator are presented.

5.) Taylor, R, M; Selcon, S, J: "Operator and Automation Capability Analysis: Picking the Right Team"

This paper provides a review of the role of operator and automation capability analysis in aircrew systems design. We chart the changing perceptions of human and machine functionality with increasing machine capability, from early pilot-in-the-loop control, through to the division and sharing of responsibilities for systems management and mission problem solving. Concepts for the integration of human and machine resources in the performance of physical and cognitive tasks, including decision-making, are discussed in the context of developments in machine intelligence. A model of cooperative teamwork, with the machine conceived of as an electronic-crew teaming resource, is proposed as broad framework for thinking about future adaptive systems requirements. We report the results of a recent study of human-electronic crew teamwork with RAF Harrier and Tornado aircrew. The results provide evidence for the validity of the teamwork model, and indicate directions for extending the capability for cooperative functioning in future aircrew adaptive systems.

6.) Eggleston, Robert, G: "Cognitive Interface Considerations for Intelligent Cockpits"

This paper presents the concept of an Intelligent Cockpit as a knowledge-based aiding system. It argues that, in order to maximally support the air crew, user aiding in two areas is required: mission task aiding and interface useability aiding. These areas of aiding are discussed in relation to four different forms of an intelligent cockpit. The central purpose of the paper, however, is to introduce the concept of a cognitive design requirement for aiding systems, and to suggest its importance to design solutions expected to achieve crew aiding in both the mission task and interface useability areas. Illustrations of possible cognitive design requirements are presented. Special attention is given to requirements that derive from human capabilities and limitations. Based on the general discussion, it is also concluded that an intelligent cockpit should be a seperate module from the traditional systems avionics, since it requires a unique process architecture.

7.) Church, T, O; Bennett II, W, S: "System Automation and Pilot-Vehicle-Interface for Unconstrained Low-Altitude Night Attack"

Unconstrained low-altitude night attack is achievable today through automation and integration of current technologies. Many of these technologies are advanced avionic systems that still require additional development before they are production-ready. However, their performance and synergistic benefits have been demonstrated. Additional efforts are still warranted to increase system safety, improve situational awareness, decrease pilot workload, and provide a more effective weapon system.

8.) Fontenilles, H de; Poutignat, P: "Evaluation automatique de combats aeriens fondee sur les intervalles caracteristiques"

The complexity of modern military simulations poses a

formidable debriefing task. A study was therefore conducted to demonstrate feasibility of an evaluation aid system for close air-to-air combat analysis. This analysis is based on a new concept of Time Interval Characterization (TIC) with breaking down a combat sequence into individual maneuvers. The TIC concept is also generic enough to ensure that the proposed aid system is adequate for every type of mission, and not only in the very specific domain of aircraft close-in combat. The analysis system contains: measurement of pilot performance, extraction of characteristical intervals, detection of good or bad behavior according to expertise rules of debriefing, optional generation of alternative trajectories by an aircraft combat expert system. Results obtained during these stages can be fully exploited with the interactive Man Machine Interface (MMI) which forms a part of the aid system. Expertise rules and MMI have been defined in consultation with relevant experts.

9.) Howard, Emily; Bitten, Robert, E: "Requirements for Pilot Assistance in a Thrust-Vectoring Combat Aircraft"

With the emergence of thrust-vectoring a/c such as the X31 and F22, new questions arise regarding the maximum potential of this technology for increasing air-to-air effectiveness. Much of this effectiveness can be attributed to the ability of the thrust-vectoring a/c to continue maneuvering while operating well beyond conventional a/c stall limits PST (post-stall maneuvering) . Comparisons with all-digital (computer-in-the-loop) simulations show that the combat effectiveness of PST is consistently greater within the all-digital analyses than within the all-manned analyses. This paper summarizes these comparisons and considers whether pilots may require supplemental assistence in order to exploit the full potential of PST ability. This paper presents the results, based upon the studies available to date. Plans for further analysis and validation studies are described at the conclusion of the paper.

10.) Urlings, P, J; Pijpers, E, W: "Overview Cockpit Technology Research and Development Programs for Improvement of the Man-Machine Interface"

This paper provides a review of the AGARD Avionics Panel (AVP) symposium on "Advanced Aircraft Interfaces: the Machine Side of the Man-Machine Interface", held in May 1992 at Madrid. The theme of this symposium was limited to the "machine-side" since a subsequent AGARD symposium at Edinburgh, Scotland, later that year was scheduled to cover the "man-side" of the subject. This paper was drafted on request of the AVP Technical Programme Committee. It summarizes the main findings of the Madrid symposium for presentation in Edinburgh. The complete text of the papers of the AVP symposium can be found in AGARD Conference Proceedings CP-521.

VI. Document Number FC 0928: Conference Proceedings

" Machine Intelligence in Air Traffic Management"

AGARD / Advisory Group for Aerospace Research and Development / Guidance and Control Panel

1993, AGARD CP-538

SUMMARY: This Symposium, namely the 56th Symposium organized by the Guidance and Control Panel of AGARD, intended to place the emphasis on the potential use of "Machine Intelligence" in the overall control loop covering all sub-loops - management, management, control, guidance, conflict alert and collision avoidance - and assessing the benefits which should or might result for the aviation community - designers and users. The response of Air Traffic Control confirmed the interest and hopes placed in advanced technologies, in particular machine intelligence. And, basic questions to be addressed include: Will Machine Intelligence solve the present difficulties of Traffic Handling? Have we any other choice? How do we evolve from the present state to a possible future fully automatic system?

RELEVANT PAPERS IN THIS REPORT:

1.) Jones, R, W: "Technical Evaluation Report"

> This volume contains the Technical Evaluation Report and the 30 papers, presented at the Guidance and Control Panel Symposium held in Berlin, Germany from 19930511 to 19930514. The papers were presented covering the following headings: Air Traffic Processes; Novel Approaches; Transition to Operation; Human/Machine Relationship; Air/Ground Integration; Phare; Ground Movements Control.

2.) Levin, Kerry, M; Fearnside, John, J: "Advances in Development Capabilities for Intelligent Air Traffic Management Systems"

> Visual presentation is a major source of information for air traffic control. Significant advances in computers, display technology, and the tools used by developers of intelligent air traffic management (ATM) systems pose challenges for the development of computer-human interface (CHIs) associated with the new automation. The CHI must be designed to be both usable and suitable. This paper reviews three capabilities available to developers of intelligent ATM systems: case-based reasoning system design, rule-based system design, and individually tailored CHI. It recommends that any intelligent ATM system be examined early in its development cycle in a laboratory environment, where it can be tested in concert with other elements of the ATM system.

3.) Bowen, David; Hlibowicki, Andrzej: "Intelligent Systems for Air Space Control and Management"

> Complete automation of an air traffic control system requires the identification of functions and their allocation to a distributed system, part of which is on the ground and part of which flies. The management of a dense cluttered air space requires a set of skills and capabilities on the part of an air traffic control team which, in some of their functions, cannot be represented algorithmically. This paper begins by presenting CompEngServ (CES) view of an automated airspace management system. The paper then presents an overview of the prototype of an advanced controller workstation developed by CES which

prototypes various portions of architecture for Airspace Management. This system represents the state of the art in ATC research. The paper then presents the future direction of research at CES and then closes with the issues which have been raised by our work that need addressing before an automated system will be practical.

4.) Planschon, P; Bonnard, M: "Use of Advanced Technologies in ATM (Air Traffic Management) Domain"

The CENA is in charge of studies related to Air Traffic Management and therefore to some of the Communication, Navigation, Surveillance means. The work is carried out to support French and European ATC in an international cooperation. It encompasses studies and experimental development aiming at operational implementation within the CAUTRA 5 programme. The different CENA projects are integrated in an experimental simulator frame ADER. This test-bed will support one of the demonstrations within PHARE, a joint European experimental programme. One division of the CENA organisation COA (Control Organisation and Automation) , deals mainly with studies aiming at providing ATM operators with helping decision tools, using advanced methods and technologies. In this paper, it can be found a short description of COA division activities and a more precise analysis of one of the project, called GOETHE, aiming to provide ATFM (Air Traffic Flow Management) regulators with a more user-friendly tool.

5.) Stengel, Robert; Wangermann, John: "Air Traffic Management as Principled Negotiation Between Intelligent Agents"

The major challenge facing the world's aircraft/airspace system (AAS) today is the need to provide increased capacity, whilst reducing delays, increasing the efficiency of flight operations, and improving safety. Technologies are emerging that should improve the performance of the system, but which could also introduce uncertainty, disputes, and inefficiency if not properly implemented. The aim of our research is to apply techniques from intelligent control theory and decision-making theory to define an Intelligent Aircraft/Airspace System (IAAS) for the year 2025. The IAAS would make effective use of the technical capabilities of all parts of the system to meet the demand for increased capacity with improved performance. This work has been supported by the FAA (Federal Aviation Agency) under FAA Grant No 92-G-0011.

6.) Hegels, Hermann, F; Hoeckstra, Willem, E: "Use of GPS (Global Positioning System) in Automated Air Traffic Control"

The Global Positioning System NAVSTAR is rapidly becoming the world standard for navigation and timing. Although primarily designed to be a military system, the civil user community is expanding at a breathtaking pace. After an introduction to the general GPS policies and the technical fundamentals this paper presents an idea on how to use GPS NAVSTAR to improve Air Traffic Control. Existing selective identification features will form the key to a GPS-based position, velocity, and acceleration message. Higher update rates and the vastly improved information on each aircraft will provide the input

for a flight plan correlation function enabling an automatic air traffic monitoring and control far beyond current standards.

7.) Jacob, Thomas; Wippich, Heinz-Georg; Schmidt, Horst; Meyer-Hilberg, Jochen; Bantle, Gerhard; Roesch, Winfried: "GPS-GNSS for ATM"

Deutsche Aerospace AG, Airborne Systems Division has developed a demonstrator to use the high accurate position data from Global Navigation Satellite Systems (GNSS) such as the US Global Positioning System (GPS) and the Russian Global Navigation System (GLONASS) . In combination with the Automatic Dependent Surveillance (ADS) function as specified by ARINC 745 the onboard computed position and flight path data is transmitted to the Aeronautical Telecommunication Network (ATN) for further use by ATM and ATC. For demonstrating ADS-functionality the onboard computed position and flight path velocity is transmitted in combination with flight management information to the ground system using a data link. All system functions have been tested and demonstrated during flight trials using a VHF data link for communication. Results of these flight tests will be presented.

8.) Gaebler, K: "ACCS (Air Command and Control System) Surveillance Exploratory Prototype (ASEP)"

Enhanced sensor and communication capabilities have given rise to increasing rates and volumes of sensor-derived and other information about air targets within modern air command and control/air traffic control (C2/ATC) surveillance systems. To help SHAPE and the NATO Air Command and Control System (ACCS) Management Agency (NACMA) to specify and implement the ACCS surveillance subsystem, the SHAPE Technical Centre (STC) is currently developing an ACCS Surveillance Exploratory Prototype (ASEP) . The purpose of the ASEP is to demonstrate the feasibility and operational benefits of future air picture generation systems. The advanced system ASEP will provide better tracking continuity and additional target information. This paper gives an overview of the following components that are implemented in the ACCS Surveillance Exploratory Prototype at STC: scenario generation and sensor simulation; real-time multisensor tracking; real-time radar data integration; external track and flight plan data integration; air picture presentation using human-computer interface (HCI) techniques.

9.) Erzberger, Heinz; Davis, Thomas, J; Green, Steven: "Design of Center-Tracon Automation System"

A system for the automated management and control of terminal area traffic, referred to as the Center-TRACON Automation System (CTAS) , is being developed at NASA Ames Research Center. This paper will review CTAS architecture, and automation functions as well as the integration of CTAS into the existing operational system. CTAS consists of three types of integrated tools that provide computer-generated advirories for both en-route and terminal area controllers to guide them in managing and controlling arrival traffic efficiently. The first component of CTAS, the Traffic Management Advisor, is being evaluated at the Denver TRACON and the Denver Air Route Traffic Control Center. The second component, the Final

Approach Spacing Tool, will be evaluated in several stages at the Dallas/Fort Worth Airport. An initial stage of the Descent Advisor tool is being prepared for testing at the Denver Center. Operational evaluations of all three integrated CTAS tools are expected to begin at the two field sites in 1995.

10.) Braven, Wim den; Bos, Hans van den: "Simulation of Fully Automated Air Traffic Control Concepts"

In order to be able to investigate various aspects of the complex Air Traffic Control (ATC) system of the future, a real-time ATC simulation facility has been constructed at NLR. The ATC automation environment of this simulator is provided by CTAS, the Center/TRACON Automation System, developed by the NASA Ames Research Center. For the simulation of air traffic, radar observations and data link, the NLR ATC Research Simulator (NARSIM) is used. The traffic samples are based on single-runway IFR operations for Schiphol Airport, with the traffic mix and distribution based on predictions for the year 2000. The results of the simulations are used to determine critical areas in ATC system automation, as well as potential benefits thereof. They can also contribute to an optimal distribution of tasks between man and machine in the ATC system of the future.

11.) Benoit, Andre; Pomeret, Jean-Marc; Swierstra, Sip: "Decision Making Aids (DMA) in On-Line ATC Systems"

This paper covers the potential of Decision Making Aids to be implemented before the year 2000 and, within the time frame considered, covers all aspects of automated assistance based on flight path prediction and monitoring, which help air traffic Controllers to establish and assess the predicted traffic situation more efficiently. Problem detection, problem minimisation, and "best next clearance" advisories will permit the reduction of the Controller's mental workload without decreasing the level of safety or his situational awareness.

12.) Beyer, R: "Considerations on Graphical User Interfaces for Intelligent ATM Support Systems"

Considerations on graphical user interfaces (GUIs) for air traffic controllers are presented in the scope of a European Air Traffic Management System (EATMS) . Fundamental issues discussed include air traffic controller tasks, human information processing and mental models, as well as automation strategies with respect to the GUI design. The more specific issues of GUI design which are discussed next include GUI programming environments and standards, development tools, design principles and human factors/human engineering standards, and usability testing. Conclusions are drawn regarding the current background of GUI design with respect to an EATMS and necessary future developments.

13.) Mahlich, S, E: "Interactive Analysis and Planning Tools for Air Traffic and Airspace Management"

Since 1989 the Institute for Flight Guidance of the German Aerospace Research Establishment (DLR) has been developing prototypes of interactive tools in close cooperation with the German Air Navigation Services (DFS) in order to achieve gradual improvements of the efficiency and productivity of the

air traffic control system. The paper briefly describes the potential of a selection of analysis and planning tools that have been developed in this framework. After an introduction into the "planning world" of tactical and strategical air traffic planning, objectives and potentials of four tools will be exemplarity demonstrated as applied to real traffic scenarios and actual problems of the current ATM system.

14.) Bichat, N; Allouche, R; Bories, A: "DAISY, a Decision Aid for an Air Situation Interpretation System"

DAISY (Decision aid for an Air situation Interpretation SYstem) developed by Alcatel ISR under a French MoD contract (DGA/DCAe/STTE procurement Agency responsible for the development of military aeronautical equipment) awarded in 1991, is a demonstrator for a tool which could support in the future the task of a military air traffic controller in the French Air Defense System STRIDA II. Concepts technical solutions and choices applied to the DAISY demonstrator presented in this paper might be applied or adapted to various air traffic surveillance problems and dual applications of this military development can be found in the civilian air traffic control field. This paper presents the DAISY demonstrator. It will address: the main functionalities; man-machine interface implementation; technical description; operational status. A conclusion will summarize benefits drawn from the approach.

15.) Adam, V; Klostermann, E; Schubert, M: "DLR's ATM Demonstration Programme"

The Institute for Flight Guidance of DLR is involved in medium and long term research and development of concepts, procedures, functions and components for a future integrated Air Traffic Management System. This paper describes a planned demonstration program which is designed to prove concepts and tools developed by the Institute in cooperation with the German ATC Authority (DFS) and the operator of Frankfurt Airport (FAG) as well as with PHARE. The aim of these experiments is to demonstrate the feasibility and merits of integration of onboard avionics with advanced ATC systems on the ground. The demonstration program will be performed in several phases comprising simulation runs in an air traffic simulator as well as flight tests with a real aircraft.

16.) Schultz, Robert, L: "Advanced Air Traffic Control and Flight Management System Concept"

A time-based air traffic control (ATC) system where vehicles are sequenced on desired time of arrival (TOA) has been proposed as one way that might help increase airport capacity. This paper evaluates three time-based ATC system concepts: ground based computing trajectories on the ground; aircraft based computing trajectories; ground and air based (generating parameterized velocity and acceleration profiles on the aircraft and transmitting them to the ground where trajectories are recomputed) . A simulation was used to evaluate the concept. The models used in the simulation are ATC trajectory generator, aircraft FMS, aircraft path controller, and vehicle motion. Multiple-aircraft scenarios, starting in cruise and descent, were examined. Factors examined were separation distances between aircraft on different approach trajectories, accuracy of TOA, and effects of wind-forecast errors on TOA

errors.

17.) Williams, David, H; Arbuckl, Douglas, P; Green, Steven, M; Braven, Wim den: "Profile Negotiation: Am Air-Ground Automation Integration Concept for Managing Arrival Traffic"

NASA Ames Research Center and NASA Langley Research Center conducted a joint simulation study to evaluate a profile negotiation process (PNP) between a time based air traffic control ATC system and an airplane equipped with a four-dimensional flight management system (4D FMS) . The Transport Systems Research Vehicle cockpit simulator was linked in realtime to the Center/TRACON Automation System (CTAS) for the experiment. Results from the experiment indicate the potential for successful incorporation to airplane preferred arrival trajectories in the CTAS automation environment. From the controllers' perspective, the main concerns were the ability of the 4D airplane to accurately track the negotiated trajectory and the workload required to support the PNP as implemented in this study.

18.) Kirstetter, B; Hunter, R, D: "Air-Ground Integration of the ATM System in PHARE"

This paper provides a general introduction into the Programme of Harmonised Air Traffic Management Research in EUROCONTROL (PHARE) . It describes the objectives of the research programme and addresses the benefits of the integration of the automated systems on board the aircraft with those of the ATC systems on the ground using a digital air-ground data link. The assumptions on the expected infrastructure an environment are explained and the possible automation and air-ground negotiation strategies discussed. Finally descriptions of the experimental facilities available or under development and of the planned experiments are provided.

19.) Adam, V; Ingle, G; Rawlings, R: "Experimental Flight Management System"

The paper reviews the requirements for an experimental Flight Management System (EFMS) and the methods adopted for its development. The functionality is described and the future application of the system is summarised. This paper is one of a group describing the aims and tasks of the Programme for Harmonised ATM Research in Eurocontrol (PHARE) . Central within the concept of PHARE is the fact that many modern a/c are fitted with FMS (Flight Management Systems) which, given the availability of accurate meteorological data, can generate and fly profiles with precision and economy. With the addition of precise time control and the ability to provide ATC with the forecast of its future profile, this FMS capability could be used to provide a significant increase in capacity whilst maintaining the present level of safety.

20.) Blom, Hen, A; Dean, Garfield; Le Guillou, Marc; Petre, Eric; Voelckers, Uwe: "The PHARE Advanced Tools"

The Programme for Harmonisation of ATM Research in Europe (PHARE) has undertaken to perform the required research work necessary for the introduction of advanced ATM. Within this PHARE framework, it is the task of the PHARE Advanced Tools (PATs) group to develop the appropriate automation and communication tools to support the air traffic controller. Although the principles for computation, prediction and control

of air traffic trajectories are well developed, the various future ATM scenarios reflect different views on the way automation and communication technology can best be applied. The consequence of this is that PHARE research has to be directed towards multiple ATM scenarios, and that the PATs to be developed should be applicable under different ATM scenarios. The paper gives an overview of the approach taken by the PATs group in facing this challenge.

21.) Velten, J, R: "The Common Modular Simulator (CMS) : An Architecture Test Bed for Future Advanced ATM Systems"

The Common Modular Simulator (CMS) project is part of the Programme for Harmonized ATM Research in Eurocontrol (PHARE) . The main objective of this project is to provide a common integration environment which shall allow to create an homogenous infrastructure in order to facilitate and harmonize the development as well as the evolution of ATM simulators in the different research establishments. To meet such an objective, CMS partners have adopted a system architecture based on a client-server model with active servers providing event subscription and event notification mechanisms. CMS will offer an architecture test bed for future advanced ATM systems. As a consequence, this project should be of great benefit to many other ATM projects.

22.) Fron, Xavier; Maudry, Bernard; Nicolaon, Jean-Pierre; Tumelin, Jean-Claude: "ARC2000 Automatic Radar Control"

The "Studies, Tests and Applied Research " (STAR) programme of the EUROCONTROL Agency is addressing several implementation timescales for ATM (Air Traffic Management) systems and procedures. ARC2000 (Automatic Radar Control 2000) is presently the major long-term component of the STAR Programme, for implementation beyond 2015. ARC2000 is addressing the En-Route ATC capacity issue, which is severe in Europe, by investigating the limit case where both major constraints, workload and sectorisation, are eleminated. ARC2000 should not be implemented as such, but should provide precious information with respect to feasible levels of automation in the long term. There are also significant by-products which will speed up shorter term research. Indeed, ARC2000 provides a 20 to 30 minute conflict-free planning which is a key feature of the European Air Traffic Management System (EATMS) concept.

23.) Moehlenkamp, Klaus; Schaenzer, Gunther: "Automatic Control Steps for Aircraft Taxi Guidance"

Modern high precision navigation systems based on satellite and inertial navigation provide a positioning accuracy that has never been achieved before, for aircraft enroute as well as during approach and on the airfield. By using such combined accurate positioning systems it is possible to guide aircraft on the ground and to perform automatic taxiing, which further increases the safety of ground operations. The new taxi guidance system GINaS, presented in this paper, is based on an integrated navigation system (DGPS/INS) and a digital map using only the standard display and navigation hardware of modern commercial aircraft. The system was successfully tested in one of our testbeds, a van. This van can be driven automatically by the system as well as by the pilot using the information of the

digital map and a drive-director. The accuracy reaches submeter level.

24.) Corrall, D, R; Clark, A, N; Hill, A, G: "Airside Ground Movements Surveillance"

There is an increasing need for surveillance, and a consequent need for automatic or semi-automatic methods for processing dynamic input data and presenting it in a form which is useful to the end user. This paper outlines advanced knowledge-based techniques for monitoring such data. The techniques have been applied to airport ground traffic applications and demonstrated in particular on data from actual turnaround scenarios for stand area servicing of an a/c as observed by a single camera. The new techniques developed can also be applied to other ground movements surveillance applications and which have multiple sensor inputs of the same or different modalities.

VII. Report: NATO Defence Research Group, Panel 8, RSG.19

"COADE - A FRAMEWORK FOR COGNITIVE ANALYSIS, DESIGN, AND EVALUATION"

1994, Final Report

Authors: P.J.M.D. Essens, J.J. Fallesen, C.A. McCann, J. Cannon-Bowers, G. Dörfel

ABSTRACT: The development of support of decision making processes in complex systems requires a systematic approach based upon knowledge of the human's role, capabilities, and the tasks to be performed. Results from a survey and a workshop show that there is a need for methodologies that systematically address the cognitive factors of complex task situations. COADE provides the developer and cognitive specialist with an approach to the development of cognitively-centred systems. The COADE framework comprises a set of activities for cognitive analysis, design and evaluation. Analysis activities result in the specification of cognitive requirements; design activities translate those into design requirements; evaluation activities control the quality of the intermediate and final products of the development process. The activities are directed towards identifying the key problems and the best opportunities for support. For each activity the framework gives a description of its purpose, product, methods and techniques, and relationships to other activities. Reference sections provide the user with background information on the activities. The remainder of the report provides further detail on models of decision making, current concepts in modelling of cognition, cognitive task analysis and knowledge elicitation techniques, an overview of known limitations and errors in cognitive performance, and a taxonomy of performance aiding strategies. A set of cognitive concepts with 'schema' as a central concept is proposed for the generic description of cognitive behaviour in decision making in Command and Control.

REPORT DOCUMENTATION PAGE

1. Recipient's Reference	2. Originator's Reference	3. Further Reference	4. Security Classification of Document
	AGARD-LS-200	ISBN 92-836-1026-1	UNCLASSIFIED/ UNLIMITED

5. Originator Advisory Group for Aerospace Research and Development
North Atlantic Treaty Organization
7 rue Ancelle, 92200 Neuilly-sur-Seine, France

6. Title

Knowledge-Based Functions in Aerospace Systems

7. Presented at/sponsored by

The Mission Systems Panel of AGARD and the Consultant and Exchange Programme of AGARD held in Madrid, Spain, on 6-7 November 1995; Châtillon, France, 9-10 November 1995; and NASA AMES Research Centre, California, USA, 16-17 November 1995.

8. Author(s)/Editor(s)	9. Date
Multiple	November 1995

10. Author's/Editor's Address	11. Pages
Multiple	156

12. Distribution Statement There are no restrictions on the distribution of this document. Information about the availability of this and other AGARD unclassified publications is given on the back cover.

13. Keywords/Descriptors

Knowledge bases	Avionics
Knowledge representation	Control equipment
Artificial intelligence	Expert computer systems
Neural nets	Automatic control
Flight control	Aircraft

14. Abstract

In aerospace systems classical control technology has enabled the transfer of functions of the human operator to machines which need not be based on the explicit evaluation of knowledge. Symbolic data processing, neural networks and the techniques of artificial intelligence now permit the design of automatic systems which can explicity make use of knowledge stored in computers.

The Lecture Series presents a conceptual framework for the automation of knowledge-based control and management functions in aerospace systems, which are usually carried out by human operators. It describes the structure of these functions, discusses successful examples of application and gives recommendations for further studies. The detailed discussion of the application examples, together with the experiences and lessons learned from these implementations will help potential builders of knowledge-based systems for aerospace applications to learn from the experts in this field.

This Lecture Series, sponsored by the Mission Systems Panel of AGARD, has been implemented by the Consultant and Exchange Programme.

AGARD

NATO OTAN

7 RUE ANCELLE • 92200 NEUILLY-SUR-SEINE

FRANCE

Télécopie (1)47.38.57.99 • Télex 610 176

DIFFUSION DES PUBLICATIONS

AGARD NON CLASSIFIEES

Aucun stock de publications n'a existé à AGARD. A partir de 1993, AGARD détiendra un stock limité des publications associées aux cycles de conférences et cours spéciaux ainsi que les AGARDographies et les rapports des groupes de travail, organisés et publiés à partir de 1993 inclus. Les demandes de renseignements doivent être adressées à AGARD par lettre ou par fax à l'adresse indiquée ci-dessus. *Veuillez ne pas téléphoner.* La diffusion initiale de toutes les publications de l'AGARD est effectuée auprès des pays membres de l'OTAN par l'intermédiaire des centres de distribution nationaux indiqués ci-dessous. Des exemplaires supplémentaires peuvent parfois être obtenus auprès de ces centres (à l'exception des Etats-Unis). Si vous souhaitez recevoir toutes les publications de l'AGARD, ou simplement celles qui concernent certains Panels, vous pouvez demander à être inclu sur la liste d'envoi de l'un de ces centres. Les publications de l'AGARD sont en vente auprès des agences indiquées ci-dessous, sous forme de photocopie ou de microfiche.

CENTRES DE DIFFUSION NATIONAUX

ALLEMAGNE
Fachinformationszentrum,
Karlsruhe
D-76344 Eggenstein-Leopoldshafen 2

BELGIQUE
Coordonnateur AGARD-VSL
Etat-major de la Force aérienne
Quartier Reine Elisabeth
Rue d'Evere, 1140 Bruxelles

CANADA
Directeur, Services d'information scientifique
Ministère de la Défense nationale
Ottawa, Ontario K1A 0K2

DANEMARK
Danish Defence Research Establishment
Ryvangs Allé 1
P.O. Box 2715
DK-2100 Copenhagen Ø

ESPAGNE
INTA (AGARD Publications)
Pintor Rosales 34
28008 Madrid

ETATS-UNIS
NASA Headquarters
Code JOB-1
Washington, D.C. 20546

FRANCE
O.N.E.R.A. (Direction)
29, Avenue de la Division Leclerc
92322 Châtillon Cedex

GRECE
Hellenic Air Force
Air War College
Scientific and Technical Library
Dekelia Air Force Base
Dekelia, Athens TGA 1010

ISLANDE
Director of Aviation
c/o Flugrad
Reykjavik

ITALIE
Aeronautica Militare
Ufficio del Delegato Nazionale all'AGARD
Aeroporto Pratica di Mare
00040 Pomezia (Roma)

LUXEMBOURG
Voir Belgique

NORVEGE
Norwegian Defence Research Establishment
Attn: Biblioteket
P.O. Box 25
N-2007 Kjeller

PAYS-BAS
Netherlands Delegation to AGARD
National Aerospace Laboratory NLR
P.O. Box 90502
1006 BM Amsterdam

PORTUGAL
Força Aérea Portuguesa
Centro de Documentação e Informação
Alfragide
2700 Amadora

ROYAUME-UNI
Defence Research Information Centre
Kentigern House
65 Brown Street
Glasgow G2 8EX

TURQUIE
Millî Savunma Başkanliği (MSB)
ARGE Dairesi Başkanliği (MSB)
06650 Bakanliklar-Ankara

Le centre de distribution national des Etats-Unis ne détient PAS de stocks des publications de l'AGARD.

D'éventuelles demandes de photocopies doivent être formulées directement auprès du NASA Center for AeroSpace Information (CASI) à l'adresse ci-dessous. Toute notification de changement d'adresse doit être fait également auprès de CASI.

AGENCES DE VENTE

NASA Center for
AeroSpace Information (CASI)
800 Elkridge Landing Road
Linthicum Heights, MD 21090-2934
Etats-Unis

ESA/Information Retrieval Service
European Space Agency
10, rue Mario Nikis
75015 Paris
France

The British Library
Document Supply Division
Boston Spa, Wetherby
West Yorkshire LS23 7BQ
Royaume-Uni

Les demandes de microfiches ou de photocopies de documents AGARD (y compris les demandes faites auprès du CASI) doivent comporter la dénomination AGARD, ainsi que le numéro de série d'AGARD (par exemple AGARD-AG-315). Des informations analogues, telles que le titre et la date de publication sont souhaitables. Veuiller noter qu'il y a lieu de spécifier AGARD-R-nnn et AGARD-AR-nnn lors de la commande des rapports AGARD et des rapports consultatifs AGARD respectivement. Des références bibliographiques complètes ainsi que des résumés des publications AGARD figurent dans les journaux suivants:

Scientific and Technical Aerospace Reports (STAR)
publié par la NASA Scientific and Technical
Information Division
NASA Headquarters (JTT)
Washington D.C. 20546
Etats-Unis

Government Reports Announcements and Index (GRA&I)
publié par le National Technical Information Service
Springfield
Virginia 22161
Etats-Unis
(accessible également en mode interactif dans la base de données bibliographiques en ligne du NTIS, et sur CD-ROM)

Imprimé par le Groupe Communication Canada
45, boul. Sacré-Cœur, Hull (Québec), Canada K1A 0S7

AGARD

NATO OTAN

7 RUE ANCELLE • 92200 NEUILLY-SUR-SEINE

FRANCE

Telefax (1)47.38.57.99 • Telex 610 176

DISTRIBUTION OF UNCLASSIFIED

AGARD PUBLICATIONS

AGARD holds limited quantities of the publications that accompanied Lecture Series and Special Courses held in 1993 or later, and of AGARDographs and Working Group reports published from 1993 onward. For details, write or send a telefax to the address given above. *Please do not telephone.*

AGARD does not hold stocks of publications that accompanied earlier Lecture Series or Courses or of any other publications. Initial distribution of all AGARD publications is made to NATO nations through the National Distribution Centres listed below. Further copies are sometimes available from these centres (except in the United States). If you have a need to receive all AGARD publications, or just those relating to one or more specific AGARD Panels, they may be willing to include you (or your organisation) on their distribution list. AGARD publications may be purchased from the Sales Agencies listed below, in photocopy or microfiche form.

NATIONAL DISTRIBUTION CENTRES

BELGIUM
Coordonnateur AGARD — VSL
Etat-major de la Force aérienne
Quartier Reine Elisabeth
Rue d'Evere, 1140 Bruxelles

CANADA
Director Scientific Information Services
Dept of National Defence
Ottawa, Ontario K1A 0K2

DENMARK
Danish Defence Research Establishment
Ryvangs Allé 1
P.O. Box 2715
DK-2100 Copenhagen Ø

FRANCE
O.N.E.R.A. (Direction)
29 Avenue de la Division Leclerc
92322 Châtillon Cedex

GERMANY
Fachinformationszentrum
Karlsruhe
D-76344 Eggenstein-Leopoldshafen 2

GREECE
Hellenic Air Force
Air War College
Scientific and Technical Library
Dekelia Air Force Base
Dekelia, Athens TGA 1010

ICELAND
Director of Aviation
c/o Flugrad
Reykjavik

ITALY
Aeronautica Militare
Ufficio del Delegato Nazionale all'AGARD
Aeroporto Pratica di Mare
00040 Pomezia (Roma)

LUXEMBOURG
See Belgium

NETHERLANDS
Netherlands Delegation to AGARD
National Aerospace Laboratory, NLR
P.O. Box 90502
1006 BM Amsterdam

NORWAY
Norwegian Defence Research Establishment
Attn: Biblioteket
P.O. Box 25
N-2007 Kjeller

PORTUGAL
Força Aérea Portuguesa
Centro de Documentação e Informação
Alfragide
2700 Amadora

SPAIN
INTA (AGARD Publications)
Pintor Rosales 34
28008 Madrid

TURKEY
Millî Savunma Başkanliği (MSB)
ARGE Dairesi Başkanliği (MSB)
06650 Bakanliklar-Ankara

UNITED KINGDOM
Defence Research Information Centre
Kentigern House
65 Brown Street
Glasgow G2 8EX

UNITED STATES
NASA Headquarters
Code JOB-1
Washington, D.C. 20546

The United States National Distribution Centre does NOT hold stocks of AGARD publications.

Applications for copies should be made direct to the NASA Center for AeroSpace Information (CASI) at the address below. Change of address requests should also go to CASI.

SALES AGENCIES

NASA Center for
AeroSpace Information (CASI)
800 Elkridge Landing Road
Linthicum Heights, MD 21090-2934
United States

ESA/Information Retrieval Service
European Space Agency
10, rue Mario Nikis
75015 Paris
France

The British Library
Document Supply Centre
Boston Spa, Wetherby
West Yorkshire LS23 7BQ
United Kingdom

Requests for microfiches or photocopies of AGARD documents (including requests to CASI) should include the word 'AGARD' and the AGARD serial number (for example AGARD-AG-315). Collateral information such as title and publication date is desirable. Note that AGARD Reports and Advisory Reports should be specified as AGARD-R-nnn and AGARD-AR-nnn, respectively. Full bibliographical references and abstracts of AGARD publications are given in the following journals:

Scientific and Technical Aerospace Reports (STAR)
published by NASA Scientific and Technical
Information Division
NASA Headquarters (JTT)
Washington D.C. 20546
United States

Government Reports Announcements and Index (GRA&I)
published by the National Technical Information Service
Springfield
Virginia 22161
United States
(also available online in the NTIS Bibliographic
Database or on CD-ROM)

Printed by Canada Communication Group
45 Sacré-Cœur Blvd., Hull (Québec), Canada K1A 0S7

ISBN 92-836-1026-1